Índice general

Índice de figuras

Índice de cuadros

Parte I

Introducción a Laboon y estudio económico de los videojuegos.

Capítulo 1

Introducción

1.1. Proyecto Propuesto

Laboon es un videojuego de *Real-Time Strategy* estrategia en tiempo real futurista de masas donde el jugador podrá controlar más de 70 unidades a la vez en feroces batallas. Con gran cantidad de edificios y unidades algunos con características únicas,enfocado a conseguir el control de los pocos recursos naturales que quedan en el planeta. Laboon está orientado a jugar partidas multijugador por internet o por red de uno contra uno aunque ofrece la posibilidad de partidas de entrenamiento para aprender a usar tus unidades y ensayar nuevas estrategias. Ofrece la posibilidad de elegir entre tres bandos; los mercenarios, los olvidados y el cónclave. Cada bando ofrecerá ventajas únicas al jugador respecto a las demás así ofreciendo distintas estrategias. Aparte del juego en sí se ofrecen funcionalidades adicionales, como crearte tu propio mapa según las indicaciones en el manual de usuario, el límite está en tu imaginación.

1.2. ¿Qué es un videojuego?

Podemos definir formalmente videojuego como un programa de ordenador creado para el entretenimiento, basado en la interacción entre una o varias personas y un aparato electrónico (ya sea un ordenador, un sistema arcade, una videoconsola, un dispositivo handheld o actualmente un teléfono móvil) que ejecuta dicho videojuego. En muchos casos, estos recrean entornos y situaciones virtuales en los que el jugador puede controlar a uno o varios personajes (o cualquier otro elemento de dicho entorno), para conseguir uno o varios objetivos por medio de unas reglas determinadas. [10]

1.2.1. Evolución de los videojuegos

Ahora hablaremos de como se concibieron los videojuegos en el mundo de las computadoras y cómo han ido evolucionando cada vez más rapido desde su aparición.

Creador de los videojuegos

Ralph Baer es considerado por muchos el inventor de los videojuegos tal como los conocemos y en su acepción más estricta, considerando que los juegos anteriores no eran aún videojuegos. En cualquier caso es el inventor de las consolas de videojuegos. Baer quería construir un sistema de videojuegos comercial para jugar en casa igual que vemos la televisión. Trabajaba en una empresa dedicada a los aparatos de televisión allá por 1951 y propuso agregar a uno de los televisores un sistema de juego interactivo, algo que resultó absurdo y fue rechazado. Posteriormente, en 1966 y por su cuenta, construyó la primera consola doméstica de videojuegos. Baer sabía lo que quería hacer pero tuvo que luchar durante años para encontrar empresas o inversores que confiaran en él para poner en el mercado su primera consola (Magnavox Odyssey), lo que por fin consiguió en 1972 con un relativo éxito. [11]

Los inicios

Durante bastante tiempo ha sido complicado señalar cuál fue el primer videojuego, principalmente debido a las múltiples definiciones de este que se han ido estableciendo, pero se puede considerar como primer videojuego el *Nought and crosses*, también llamado *OXO*, desarrollado en 1952. El juego era una versión computarizada del tres en raya que se ejecutaba sobre la EDSAC (el primer calculador electrónico del mundo) y permitía enfrentar a un jugador humano contra la máquina. [10]

Figura 1.1: Primer Videojuego: OXO.

En 1958 William Higginbotham creó, sirviéndose de un programa para el cálculo de trayectorias y un osciloscopio, *Tennis for Two*. Un simulador de tenis de mesa para entretenimiento de los visitantes del Brookhaven National Laboratory. Este videojuego fue el primero en permitir el juego entre dos jugadores humanos. Cuatro años más tarde Steve Russell, un estudiante del Instituto de

Tecnología de Massachussets, dedicó seis meses a crear un juego para computadora usando gráficos vectoriales: *Spacewar!*. En este juego, dos jugadores controlaban la dirección y la velocidad de dos naves espaciales que luchaban entre ellas. El videojuego funcionaba sobre un PDP-1 y fue el primero en tener un cierto éxito aunque apenas fue conocido fuera del ámbito universitario. [10]

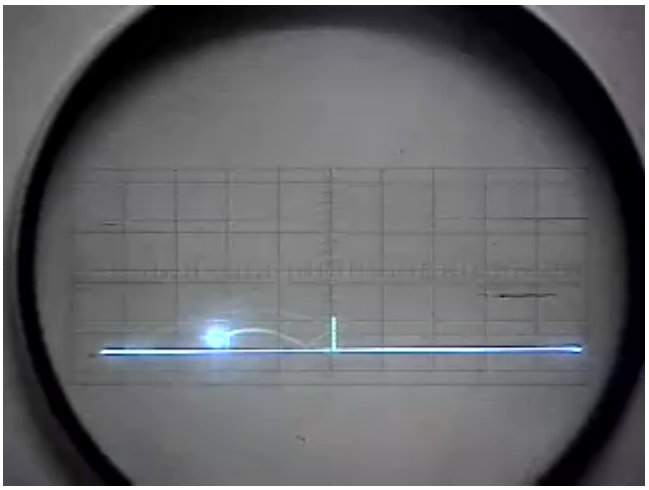

Figura 1.2: Tennis For Two.

En 1966 Ralph Baer empezó a desarrollar junto a Albert Maricon y Ted Dabney un proyecto de videojuego llamado *Fox and Hounds* dando inicio al videojuego doméstico. Este proyecto evolucionaría hasta convertirse en la Magnavox Odyssey, la primer consola doméstica de videojuegos lanzado en 1972 que se conectaba a la televisión y que permitía jugar a varios juegos pregrabados. La Odyssey Home Sistem Entertenement es considerada la primer generación de videoconsolas. [10]

Años 1970: La eclosión de los videojuegos

Un hito importante en el inicio de los videojuegos tuvo lugar en 1971 cuando Nolan Bushnell comenzó a comercializar *Computer Space*, una versión de *Space War*, en Estados Unidos. [10]

Figura 1.3: Computer Space.

La ascensión de los videojuegos llegó con la máquina recreativa *Pong*, muy similar al *Tennis for Two* pero utilizada en lugares públicos: bares, salones, etc. El sistema fue diseñado para Nolan Bushnell en la recién fundada **Atari**. [10]

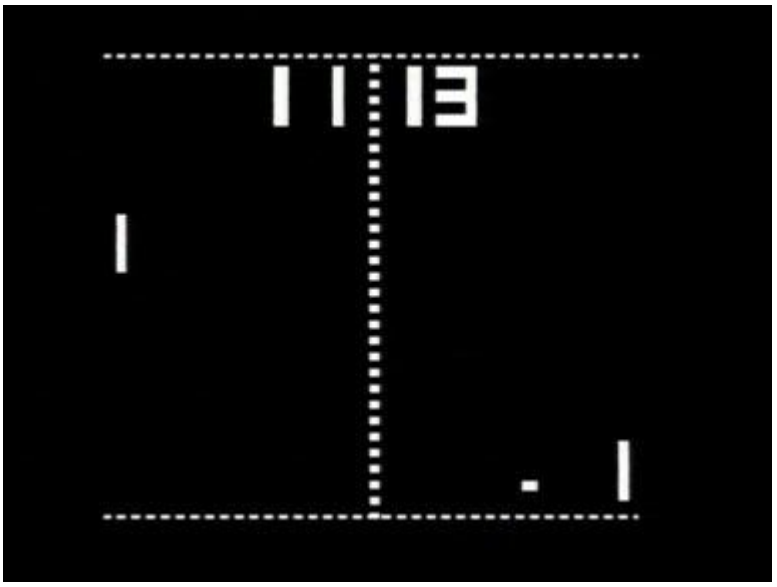

Figura 1.4: Pong.

El juego se presentó en 1972 y fue la piedra angular del videojuego como industria. Durante los años siguientes se implantaron numerosos avances técnicos en los videojuegos (destacando los microprocesadores y los chips de memoria), aparecieron en los salones recreativos juegos como *Space Invaders* o *Asteroids* y sistemas domésticos como el Atari 2600. [10]

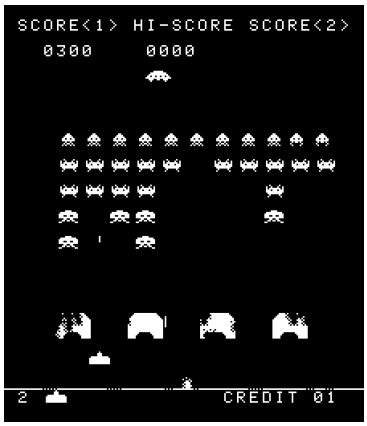

Figura 1.5: Space Invaders.

Años 1980: La década de los 8 bits

Los años 80 comenzaron con un fuerte crecimiento en el sector del videojuego alentado por la popularidad de los salones de máquinas recreativas y de las primeras videoconsolas aparecidas durante la década de los 70. [10]

El negocio asociado a esta nueva industria alcanzó en poco tiempo grandes cosas. Sin embargo, en 1983 comenzó la que se ha dado por llamar crisis del videojuego de 1983, la cual afectó principalmente a Estados Unidos y Canadá, y que no llegaría a su fin hasta 1985. [10]

En 1985 apareció Super Mario bros. que supuso un punto de inflexión en el desarrollo de los juegos electrónicos. La mayoría de los juegos anteriores sólo contenían unas pocas pantallas que se repetían en un bucle y el objetivo simplemente era hacer una alta puntuación. El juego desarrollado por Nintendo supuso un estallido de creatividad. Por primera vez teníamos un objetivo y un final en un videojuego. En los años posteriores otras compañías emularon su estilo de juego. [10]

Otra rama de los videojuegos que creció con fuerza fue la de los videojuegos portátiles. Estos comenzaron a principios de los 70 con los primeros juegos completamente electrónicos lanzados por Mattel, los cuales difícilmente podían considerarse como videojuegos, y fueron creciendo en popularidad gracias a conversiones de recreativas como las realizadas por Coleco o adictivos micro juegos como las Game & Watch de Nintendo. La evolución definitiva de las portátiles como plataformas de videojuego llegó en 1989 con el lanzamiento de la Game Boy (Nintendo). [10]

Figura 1.6: Primera consola portátil de Nintendo: Gameboy.

Años 1990: La revolución del 3D

A principios de los años 90 las videoconsolas dieron un importante salto técnico gracias a la competición de la llamada "generación de 16 bits" compuesta por Mega Drive (Sony) y Super Nintendo (Nintendo), así como Turbografx (NEC). [10]

Esta generación supuso un importante aumento en la cantidad de jugadores y la introducción de tecnologías como el CD-ROM, además de una importante evolución dentro de los diferentes géneros de videojuegos, principalmente gracias a las nuevas capacidades técnicas. [10]

Mientras tanto diversas compañías habían comenzado a trabajar en videojuegos con entornos tridimensionales, principalmente en el campo de los PC, obteniendo diferentes resultados desde las "2D y media" de *Doom*, 3D completas de *4D Boxing* a las 3D sobre entornos pre-renderizados de *Alone in the Dark*. [10]

Figura 1.7: Primer Videojuego con captura de movimientos reales: 4DBoxing.

Figura 1.8: Alone In The Dark.

Por su parte los arcades comenzaron un lento pero imparable declive según aumentaba el acceso a consolas y ordenadores más potentes. Para intentar compensar la huida de clientes, los fabricantes de máquinas arcade apostaron por potenciar hardwares específicos que difícilmente podían copiarse en un sistema doméstico como coches de tamaño real o pistas de baile entre otros. [10]

Desafortunadamente la gran inversión que suponían estos aparatos sacó del mercado a muchos recreativos, con lo que el arcade pasó de ser un entretenimiento popular a estar recluido en unos pocos lugares muy específicos. [10]

En PC eran muy populares los FPS (First Person Shooter) como *Quake* o *Half-Life* (Valve) y los RTS (Real Time Strategy) como *Command & Conquer* o Starcraft. Además las conexiones entre ordenadores mediante internet facilitaron el juego multijugador, convirtiéndolo en la opción predilecta de muchos jugadores, y fueron las responsables del nacimiento de los MMORPG como Ultima Online (Origin). Finalmente en 1998 apareció en Japón la Dreamcast (Sega), la cual llegaría a occidente en 1999 y daría comienzo a la "generación de los 128 bits". [10]

Figura 1.9: FPS: Half-Life.

Figura 1.10: RTS: StarCraft.

Años 2000: El comienzo del nuevo siglo

Entre los años 2000 y 2001 se produjo una pelea entre "titanes" a nivel de consolas. Sony había lanzado en 2000 su consola estrella, PlayStation 2 mientras que por otro lado Microsoft en 2001 decidió entrar en la industria de los videojuegos creando la XBOX.

En 2005 estos dos gigantes continuaron su lucha con la XBOX 360 (Microsoft)y en 2006 con PlayStation 3 (Sony) incorporando cada una un enfoque nuevo a las distintas tecnologías que se están abriendo paso en el mercado a pasos agigantados.

Por otra parte el ordenador personal es la plataforma más cara de juegos pero también la que permite mayor flexibilidad. Esta flexibilidad proviene del hecho de poder añadir al ordenador componentes que se pueden mejorar constantemente, como son tarjetas gráficas, de sonido, accesorios como volantes, pedales y mandos, etc. Además es posible actualizar los juegos con parches oficiales o con nuevos añadidos realizados por la compañía que creó el juego o por otros usuarios. [10]

1.3. Clasificación del género de los videojuegos

Un género de videojuego designa un conjunto de videojuegos que poseen una serie de elementos comunes. A lo largo de la historia de los videojuegos aquellos elementos que han compartido varios juegos han servido para clasificar como un género aquellos que les han seguido, de la misma manera que ha pasado con la música o el cine. [10]

Los videojuegos se pueden clasificar como un género u otro dependiendo de su representación gráfica, el tipo de interacción entre el jugador y la máquina, la

ambientación y su sistema de juego, siendo este último el criterio más habitual a tener en cuenta. [10]

Hay que decir que cada vez es más es habitual que un juego contenga elementos de diversos géneros, cosa que hace difícil su clasificación. [10]

He aquí una posible clasificación con una breve descripción de cada uno de los elementos:

1.3.1. Beat 'Em Up

Los beat'em up también llamados "Yo contra el barrio" son juegos similares a los de lucha, con la diferencia de que en este caso los jugadores deben combatir con un gran número de individuos mientras avanzan a lo largo de varios niveles. En los beat'em up suele ser posible jugar dos o más personas a la vez de forma cooperativa para facilitar el progreso. Un juego típico de este género es *Streets of Rage*. [10]

Figura 1.11: Beat 'Em Up: Streets of Rage 3

1.3.2. Lucha

Los juegos de lucha, recrean combates entre personajes controlados tanto por un jugador como por la computadora. El jugador ve a los combatientes desde una perspectiva lateral, como si se tratase de un espectador. Este tipo de juegos ponen especial énfasis en las artes marciales, reales o ficticias, u otros tipos de enfrentamientos sin armas como el boxeo o la lucha libre. Otros juegos permiten también usar armas blancas como pueden ser espadas, hachas, martillos, etc. o ataques a distancia, normalmente de carácter mágico o etéreo. [10] Hay muchos juegos que encajan perfectamente en este contexto: *Street Fighter, Fatal Fury, Mortal Kombat, Tekken, Soul Calibur*, etc.

Figura 1.12: Lucha: Tekken 6

1.3.3. Shooter

Primera Persona

En los juegos de Disparos en primera persona (FPS), las acciones básicas de este tipo de juegos son mover al personaje y usar un arma, un arma se anuncia en la pantalla en primer plano y el jugador puede interactuar con éste. Esta perspectiva tiene por meta dar la impresión de estar detrás del personaje. [10]

Generalmente en estos juegos la calidad del guión no se trabaja mucho (salvo superproducciones), mientras que hacen destacar la calidad gráfica y la jugabilidad. Cada generación de juegos mejora con las últimas tecnologías y las nuevas posibilidades gráficas que rozan el fotorrealismo. La mecánica del juego impone generalmente al jugador tener buenos reflejos y precisión. [10]

Algunos videojuegos marcaron el género, tales como *Doom, Half Life, Unreal, Quake, Halo.* [10]

Es de notar que ciertos juegos de acción en primera persona solo pueden jugarse en modo Multijugador a través del uso de Internet.

Figura 1.13: FPS: Quake 4

Tercera Persona

Los juegos de disparos en tercera persona (TPS) se basan en alterner entre disparos y pelea o interacción con el entorno, pero a diferencia de los FPS, se juega con un personaje visto desde atrás y en ocasiones, desde una perspectiva isométrica. [10]

Estos juegos sacrifican la precisión a la ganancia de una gran libertad de movimientos. [10]

Dentro de esta clasificación, han tomado fuerza un nuevo grupo de subgénero, el cual a pesar de seguir una historia, da un gran énfasis en permitirte moverte y hacer lo que tú quieras, el más representativo y pélemico de este subgénero seria sin duda Grand Theft Auto, donde las posibilidades con el personaje son casi infinitas. Aunque no hay todavía un nombre con el que la industria se ponga de acuerdo, la subcategoría se ha denominado 'sandbox', aludiendo a que las posibilidades de moldear lo que quieres hacer es similar a una caja de arena. [10]

Los juegos como la saga *Grand Theft Auto* o *Prototype* son algunos ejemplos aunque hay muchos otros ejemplos.

Figura 1.14: TPS: Prototype

Shoot 'em Up

Este género de juegos de disparos está basado en el continuo uso de un arma, con la diferencia de las otras secciones de que pasas la mayor parte del tiempo disparando, con frecuencia la pantalla esta repleta de balas, tanto propias como del enemigo. Es común en este género distintos tipos de armas, o la mejora de las disponibles, avance automático o lateral, y enfrentamientos con jefes enormes al final de cada misión, los cuales en la mayoría de los casos, ocupan más de la mitad de la pantalla. Aunque el representante de este género podría ser el juego de los 80's *Contra*, tambien los juegos de naves tipo *Gradius* o *r-Type* son dignos exponentes del genéro shoot'em up. [10]

Aún en estos tiempos, se prefiere que la acción se desarrolle en 2D, aunque se incluyen o combinan elementos 3D para dar efectos de profundidad, explosiones o mayor efecto visual, sobre todo con los jefes. [10]

Figura 1.15: Shoot 'Em Up: Metal Slug 3

1.3.4. Infiltración

Los juegos de infiltración son un género relativamente reciente. Estos juegos se basan en el sigilo, la furtividad y la estrategia en vez de buscar la confrontación directa con el enemigo. [10]

Normalmente los juegos de infiltración aparecen como un subgénero de los juegos de disparos, aunque podemos encontrar juegos como *Commandos*, que se puede clasificar a la vez como juego de estrategia y de infiltración. [10]

Ejemplos: *Metal Gear, Splinter Cell, Hitman, Thief, Beyond Good & Evil.*

Figura 1.16: Infiltración: Hitman 2

1.3.5. Plataformas

En los juegos de plataformas el jugador controla a un personaje que debe avanzar por el escenario evitando obstáculos físico, ya sea saltando, escalando o agachándose. Además de las capacidades de desplazamiento como saltar o correr, los personajes de los juegos de plataformas poseen frecuentemente la habilidad de realizar ataques que les permiten vencer a sus enemigos, convirtiéndose así en juegos de acción. Inicialmente los personajes se movían por niveles con un desarrollo horizontal, pero con la llegada de los gráficos 3D este desarrollo se ha ampliado hacia todas las direcciones posibles. [10]

A partir del sistema de cambio de perspectiva utilizado en el famoso *Super Mario 64* se ha permitido una verdadera libertad de movimiento en tales ambientes. [10]

Los juegos de plataformas son uno de los primeros tipos de juegos que aparecieron en los ordenadores. Aunque este género fue muy popular en los 80 y los 90, su popularidad ha disminuido en los últimos años, aún más cuando se introdujeron los gráficos 3D en los videojuegos. Esto se debe en gran parte a que

las 3D han hecho que se perdiese la simplicidad de desarrollo que caracterizaba este género. [10]

Ejemplos: *Super Mario, Sonic the Hedgehog, Alex Kidd, Crash Bandicoot, Rayman, Spyro the Dragon, Donkey Kong, Tomb Raider.*

Figura 1.17: Plataformas: Super Paper Mario

1.3.6. Musicales

Este tipo de juegos consiste en representar a uno de las personas integrantes de una banda. Podemos representar a un cantante o a los distintos componentes del grupo (bajista, batería, teclado...). Dentro de este ámbito también podemos encontrar los juegos donde uno tiene que bailar al ritmo de una canción de una forma determinada.

Los principales juegos dentro de este ámbito son: *Singstar, Dance Dance Revolution.*

Figura 1.18: Musicales: Singstar

1.3.7. Simulación de Combate

Género poco llevado a la práctica que se caracteriza por el elevado realismo en todos los aspectos relevantes en cuanto al desarrollo de las partidas. El máximo exponente de este subgénero lo encontramos en *Operation Flashpoint*.En este tipo de juegos en los que un solo pero certero disparo significa la muerte, el movimiento de los personajes o el comportamiento del armamento tratan de ser absolutamente realistas. [10]

Figura 1.19: Simulador de Combate: Operation FlashPoint

1.3.8. Simulación de Construcción

Genero muy popular en PC, donde el programa le proporciona al usuario todas las herramientas para construir un proyecto, el cual debe ser lo más apegado a la realidad como sea posible, en el cual se consideran desde gastos de construcción y mantenimiento, hasta una línea de tiempo, física y clima que afecta todas las decisiones que tomes. La particularidad de poder experimentar, tomar decisiones y afectar el desempeño de tu simulación, los hace tremendamente adictivos. Juegos desde simulación de jardines hasta construcciones de casas, parques y ciudades. Los más populares sin duda son *Sim City*, en PC, donde el usuario toma el rol de alcalde y debe administrar una ciudad. [10]

Figura 1.20: Simulador de Construcción: Sim City

1.3.9. Simulación de Vida

Tomando el mismo concepto del género, el simulador de vida se enfoca en controlar un personaje con capacidades y emociones humanas, y controlar todos los aspectos de su vida, desde donde vivirá, que estudiará, con quien se casará. El realismo colocado en estos juegos y una línea de tiempo que permite al personaje evolucionar, comer, dormir, convivir con otras personas, envejecer e incluso morir, hace que muchas personas tomen estos juegos como desahogo de todo lo que quisieran experimentar en su vida propia, pero sin riesgos. El más popular y culpable de definir este género es el juego de *The Sims*, además del juego en sí, en PC se han lanzado muchas expansiones, donde se permite incorporar al juego principal nuevas posibilidades en varios ámbitos de la vida cotidiana, o no tan cotidiana. La evolución de este juego llegó el año pasado con *Spore*, un juego donde comienzas siendo una bacteria, y dependiendo de las decisiones tomadas, puedes evolucionar a tu criatura de formas casi infinitas. [10]

Figura 1.21: Simulador de Vida: Spore

1.3.10. Arcade

Los juegos de arcade, se caracterizan por la simplicidad de acción rápida de jugabilidad, esto obtuvo la gloria en la época de 1980. No requiere historia, solo juegos largos o repetitivos. *Space Invaders, Asteroids, Pac-Man, Missile Command, Galaxian* son ejemplos notables de arcade. [10]

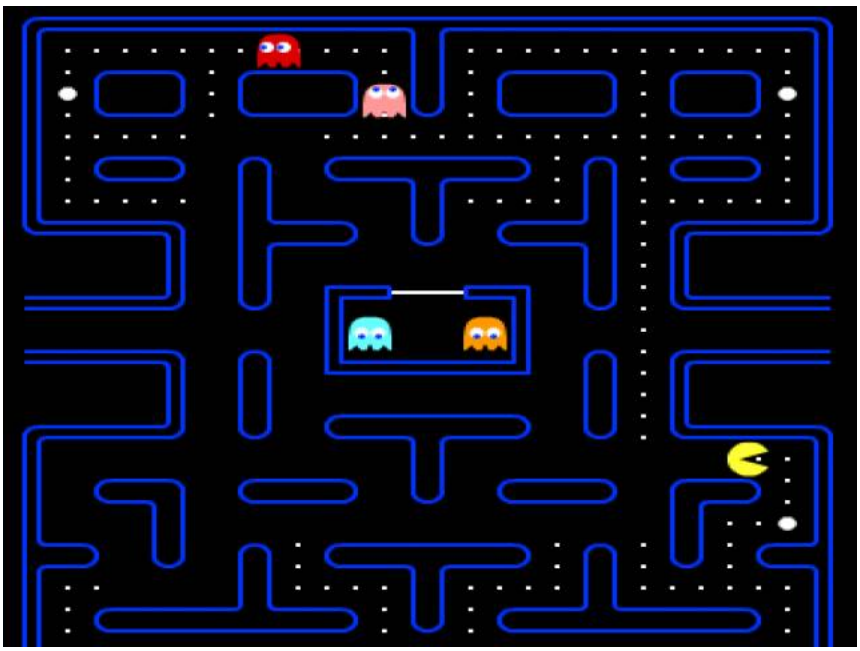

Figura 1.22: Arcade: Pac-Man

1.3.11. Deportes

Sport

Los juegos de Deporte son aquellos que simulan juegos de deporte real, entre ellos encontramos, golf, tenis, futbol, hockey, juegos olímpicos, etc. Una gran mayoría entre ellos, notablemente *Tony Hawk's Pro Skater*. El participante directamente lo juega a través del control. El propósito es el mismo que el deporte original, aunque puede contener variantes. [10]

Figura 1.23: Sport: Pro Evolution Soccer 5

Carreras

Principalmente son juegos que se dedican a comenzar de un punto y llegar a una meta antes que los contrincantes. Juegos así se han desarrollado de su forma común en vehículos, hasta otras formas como juego de plataformas. La idea principalmente es competir en llegar primero, y algunas veces se suelte ampliar este concepto, originando herramientas y trampas para la carrera. [10]

Los simuladores de carreras representan con la exactitud las carreras de la actualidad, seguido por variaciones en detalles y agregados. [10]

Dentro de los juegos orientados a la simulación tenemos: *Gran Turismo, Toca Touring Car, GTR, Forza Motorsport, Colin McRae Rally.* [10]

Y sus posibles variantes: *Sonic Riders, Mario Kart, Burnout, Out Run, F-Zero, Mach Rider.*

Figura 1.24: Carreras Variante: Super Mario Kart Wii

Figura 1.25: Carreras Simulación: Gran Turismo 5

1.3.12. Agilidad Mental

Estos son juegos donde tienes que pensar y agilizar el pensamiento. El objetivo aquí es resolver ejercicios con dificultad progresiva para desarrollar la habilidad mental. [10]

Juegos como estos son: *Brain Training, Brain Academy, Tetris, buzz.* [10]

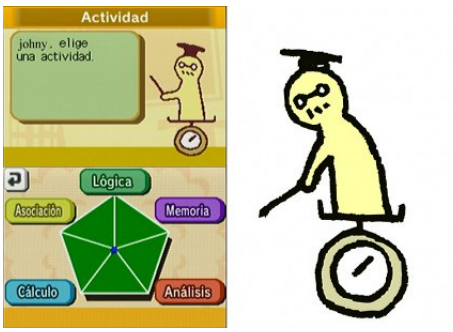

Figura 1.26: Agilidad Mental: Brain Academy

1.3.13. Educación

Aunque antiguamente solo se ha usado para juegos infantiles, los juegos educativos son aquellos que enseñan mientras promueven diversión o entretenimiento (Didactismo). A diferencia de una enciclopedia, trata de entretener mientras se memoriza conceptos o información. En algunos casos se duda de que sea un género de video juego, ya que el concepto no está muy desarrollado, ya que en cierto caso estos juegos fallan enseñando más que leer y aprender que darlo a la práctica. [10]

Juegos como estos son: Antiguos juegos infantiles: *Fine Artist, El autobús mágico.* [10]

1.3.14. Aventura Clásica

Los juegos de aventura fueron, en cierto modo, los primeros videojuegos que se vendieron en el mercado, empezando por *Colossal Cave Adventure* en los años 1970. Este tipo de juego se hizo famoso con los juegos de la serie *Zork* y consiguió alcanzar cierto nivel de popularidad en los años 80 que duró hasta mediados de los 90. El jugador encarna a un protagonista que por lo general debe resolver incógnitas y rompecabezas con objetos diversos. Los primeros videojuegos de aventura eran textuales (aventuras textuales, aventura conversacional o ficción interactiva). En estos, el jugador utiliza el teclado para introducir órdenes como “coger la cuerda” o “ir hacia el oeste” y el ordenador describe lo que pasa. Cuando el uso de gráficos se generalizó, los juegos de aventura textuales dejaron paso a los visuales (por ejemplo, con imágenes del lugar presente) que substituyeron de este modo las descripciones por texto, que se habían vuelto casi superfluas. Estos juegos de aventura con gráficos seguían, no

obstante, sirviéndose de la introducción de texto. Además, sigue existiendo una comunidad de autores y jugadores activos de ficción interactiva (véase la web del CAAD para el mundo hispanoparlante), aunque la participación de grandes empresas comerciales es más bien rara. [10]

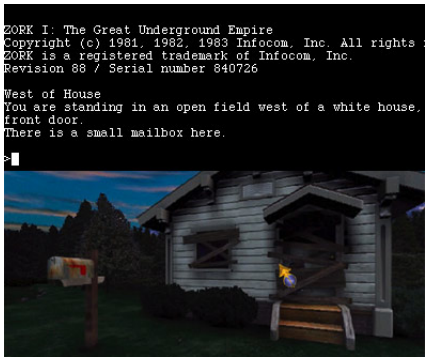

Figura 1.27: Aventura Conversacional: Zork

1.3.15. Aventura Gráfica

A comienzos de los 1990, el uso creciente del ratón dio pie a los juegos de aventura de tipo "Point and Click", también llamados aventura gráfica, en los que ya no se hacía necesaria la introducción de comandos. El jugador puede, por ejemplo, hacer clic con el puntero sobre una cuerda para recogerla. [10]

A finales de los 1990, este tipo de juego sufrió una importante pérdida de popularidad, las presentaciones de productos de masas se hicieron raras, hasta el punto que hubo quienes predijeron la muerte de este tipo de videojuegos. No obstante, en 2005, los juegos de aventura experimentaron un retorno importante. Los grandes juegos de aventura de la historia incluyen títulos como *Day of the Tentacle*, los juegos de la serie *King's Quest*, la serie *Leisure Suit Larry*, la serie *Broken Sword*, la serie *Gabriel Knight*, la serie *Police Quest*, la serie *Space Quest* y los de *Monkey Island*. [10]

Figura 1.28: Aventura Gráfica: Monkey Island 1

1.3.16. Juego de Rol (RPG)

De las siglas en ingles Role Playing Game, se caracterizan por la interacción con el personaje, una historia profunda y una evolución del personaje a medida que la historia avanza. Para lograr la evolución generalmente se hace que el jugador se enfrasque en una aventura donde ira conociendo nuevos personajes, explorando el mundo para ir juntando armas, experiencia, aliados e incluso magia. [10]

Aunque la mayoría de juegos de aventura incluyen una dosis baja de RPG, los RPG puros se enfocan específicamente en subir experiencia y personalización del personaje, en juegos como *The Elder Scroll IV*, el crear y personalizar tu personaje puede llevar hasta 30 horas. Los RPG clásicos, inspirados en los juegos de tablero, realizan las batallas por turnos, es decir, el jugador usa su equipo y habilidades aprendidas para atacar mediante una serie de comandos y después debe quedar estático y esperar a recibir el ataque del otro jugador o CPU. el mejor ejemplo de esto es *Final Fantasy, Dungeons & Dragons.* Aunque aún hay RPGs clásicos, la mayoría ahora usan el combate en tiempo real, es decir, no hay pausas y ambos atacan al mismo tiempo. [10]

En otra subcategoría de los RPG se han puesto de moda los RPG en línea o MMORPG (Massive Multiplayer Online RPG)donde cada jugador crea un personaje y mediante una conexión a internet, entra a un mundo donde miles de jugadores se unen a la aventura, exploran, intercambian y evolucionan juntos. El juego con mayor auge es sin duda *World of Warcraft*, lo mismo que *Everquest*, donde la adicción a estar dentro del juego ha causado desde aislamientos y despidos de gente que abandona sus actividades con tal de seguir jugando, hasta personas que se conocen en el juego y terminan casándose en la vida real. [10]

Figura 1.29: MMORPG: World of Warcraft

1.3.17. Juegos de Estrategia

Estrategia por turnos

También se los conoce por sus siglas en inglés TBS (turn-based strategy). El término "juego de estrategia por turnos" generalmente se aplica a ciertos videojuegos de estrategia para distinguirlos de los juegos de estrategia en tiempo real. Un jugador de un juego por turnos posee un período de análisis antes de realizar una acción. Algunos ejemplos de este género son *Civilization*, y las sagas *Heroes of Might and Magic* y *Master of Orion*. [10]

Los juegos por turnos vienen en dos formas dependiendo de si, en un turno, los jugadores juegan simultáneamente o juegan sus turnos en secuencia. Los primeros son llamados juegos de estrategia por turnos simultáneos; siendo Diplomacy un ejemplo notable. Los últimos caen en la categoría de los juegos de estrategia por turnos alternados, y a su vez se subdividen en:

- Posicional
- Inicio round robin
- Aleatorio

La diferencia está en el orden en el que los jugadores juegan sus turnos. En posicional, los jugadores empiezan sus turnos en el mismo orden siempre. En el inicio round robin, el jugador que empieza es elegido de acuerdo a una política round robin. En aleatorio el primer jugador es elegido al azar. [10]

Casi todos los juegos de estrategia que no sean videojuegos (juegos de mesa) son por turnos. El mercado de los juegos de computadora últimamente se ha inclinado más por los juegos en tiempo real. [10]

Algunos juegos recientes han mezclado componentes de los juegos en tiempo real con componentes de los juegos por turnos. En estos juegos a los jugadores se les proporcionan dados para 100 movimientos por día. Estos movimientos pueden ser tomados en cualquier momento de ese día independientemente de si otros jugadores hayan realizado sus movimientos o no. [10]

Figura 1.30: TBS: Heroes of Might And Magic V

Estrategia en tiempo real

También conocidos por sus siglas en inglés RTS (real-time strategy). Generalmente aplicado solo a ciertos juegos de estrategia de computadora, el nombre "juego de estrategia en tiempo real" indica que la acción en el juego es continua, y los jugadores tendrán que tomar sus decisiones en un estado de juego constantemente cambiante. La jugabilidad de estos tipos de juegos se caracteriza por la obtención de recursos, construcción de bases, investigación de tecnologías y producción de unidades. [10]

Algunos jugadores discuten la importancia de la estrategia en los juegos de estrategia en tiempo real, ya que la habilidad y destreza manual son vistas como los factores claves en este género de juegos. Trony Dunniway dijo: "Un jugador controla cientos de unidades, docenas de edificios y muchos eventos que pasan todos al mismo tiempo. Solo hay un jugador, y él solo puede prestar atención a una sola cosa a la vez. Los jugadores expertos pueden cambiar rápidamente entre muchas tareas distintas, sin embargo los jugadores casuales tienen más problemas con esto". Ernest Adams incluso sugiere que la jugabilidad en tiempo real interfiere con la estrategia. "El pensamiento estratégico, por lo menos en el campo de la jugabilidad, no se lleva muy bien con la acción en tiempo real". [10]

Dune II, creado por Westwood Studios, es considerado el padre de los RTS. Este fue seguido por la saga *Command & Conquer*. Otros juegos de estrategia en tiempo real populares son *Total Annihilation* de Cavedog, la saga *Warcraft* de Blizzard Entertainment, *Starcraft* (1998), y la saga *Age of Empires* de Ensemble Studios. [10]

Figura 1.31: RTS: WarCraft III

1.4. Importancia de los videojuegos hoy en día

Los videojuegos son comúnmente catalogados como un "medio de entretenimiento", así como usualmente se cataloga al cine, a la televisión y al radio, esto se explica por la forma y el contenido del videojuego comercial general, ya que durante el principio del boom de la industria, esta era una tecnología enfocada principalmente para los niños, algo que ha cambiado con las nuevas generaciones. Decir hoy en día que los videojuegos son "entretenimiento" es anticuado porque no se está pensando y poniendo atención a la real y visible potencia-aplicación de este medio, limitando así las interpretaciones y percepciones que se le pueda atribuir al videojuego y por lo tanto su nivel de valor. [12]

El videojuego, como todo medio de comunicación, es un producto cultural que corresponde a un contexto, a una sociedad y a unos fines, también cambia, desde su capacidad como tecnología hasta la capacidad de contenido, en un proceso que generalmente está constituido por personas de muy diferentes disciplinas (programadores, diseñadores gráficos, escritores, etc.) y por esto ha sido revalorizado durante la última década. [12]

Fue un fenómeno tecnológico en su inicio y luego se convirtió en un fenómeno social, ya que muchos niños y jóvenes se han vuelto "videojugadores " desde aquellos primeros años a principio de los '70, y ahora esas personas y los niños de los últimos 30 años han hecho crecer a la industria del videojuego de una manera impresionante. [12]

El videojuego es hoy una industria cultural y como tal en ella se encuentran representadas las necesidades, ilusiones, fantasías, gustos, intelecto y capacidad de narración, entre otras, de la sociedad de principio del siglo XXI. Pero el videojuego, como todo objeto cultural, no solamente es producto de la cultura y sociedad sino también el mismo hecho de existir la cambia y puede llegar a transformarla. Hay que poner mucha atención a este punto ya que el videojuego en muchos aspectos, por ejemplo: comunidades en línea, están cambiando a la sociedad y cultura que los creo e hizo crecer. El juego de video tiene muchas aplicaciones en un sentido práctico, una de ellas es su capacidad educativa en el usuario. El videojuego es parte de la revolución digital que vive nuestra sociedad actualmente. Este seguramente será el medio de entretenimiento que más se desarrolle y explote en el inicio de este siglo XXI. Su papel en los medios de comunicación apenas está empezando. [12]

La industria que se ha creado en torno a los videojuegos mueve miles de millones cada año, y cada año que pasa tiende a incrementar los ingresos generados por la misma.

Estos ingresos fueron superiores a 18.000 millones de dólares a nivel mundial en el año 2006 mientras que en el año 2007 los ingresos superaron los 23.000 millones de dólares. El siguiente cuadro muestra un análisis de los ingresos cosechados por 6 grandes empresas que se dedican a desarrollar videojuegos en 2006 y 2007.

Empresas	2006	2007
Sony Games	7.988,3	8.765,5
Nintendo Co. Ltd. (Hardware)	2.345,7	5.041,4
Nintendo Co. Ltd. (Software)	1.879,9	3.272,2
Electronics Arts Inc.	2.951,0	3.091,0
Activision Inc.	1.468,0	1.513,0
THQ Inc.	806,6	1.026,9
Ubisoft	662,0	870,8
Total	**18.101,5**	**23.580,8**

Cuadro 1.1: Ingresos en millones de dólares de distintas empresas [13]

Como podemos observar, existe un incremento considerable de los ingresos de estas empresas, en algunos casos el incremento es más que considerable. El incremento de *Ninctendo Co. Ltd. (Hardware)* fue de un 114.92 %, esto refleja como esta industria se ha abierto paso en el mundo de los negocios a pasos agigantados.

Otra característica importante es que cada año se lanzan de a miles de títulos a la venta, pero el mercado se encuentra dominado por un número reducido de juegos que constituyen grandes éxitos denominados *hits*. [13]

Mencionado el párrafo anterior podemos hacer una referencia concreta a un juego con nombre y apellido. En septiembre de 2008 se lanzó para la consola XBOX 360 el juego *Halo 3*, el cual recaudó en el día de su lanzamiento la friolera de 170 millones de dólares, convirtiéndose en el lanzamiento más exitoso de todos los tiempos. En la primera semana en el mercado las ventas alcanzaron los 300 millones de dólares. [13]

Figura 1.32: Halo3 - XBOX360

Pero claro no sólo se generan ingresos en esta industria por el desarrollo de videojuegos. Las videoconsolas son unas grandes fuentes de ingresos para las compañías que las desarrollan. Si nos centramos en las marcas Sony, Microsoft y Nintendo podemos observar que a nivel mundial el número de consolas vendidas puede parecer incluso desorbitado.

Compañía	Consola	Consolas Vendidas	Precio	Ingresos
Sony	PlayStation 3	19.5 Mill	399€	7780.5 Mill €
Nintendo	Nintendo Wii	45 Mill	250€	11250 Mill €
Microsoft	XBOX360	28 Mill	300€	8400 Mill €
				Ingresos Totales: 27430.5 Mill €

Cuadro 1.2: Consolas vendidas por Sony, Nintendo y Microsoft [14]

De todo esto podemos concluir que la industria de los videojuegos se ha convertido en una de las más importantes y lucrativas a nivel mundial y sigue creciendo a un ritmo casi temerario, no sólo en beneficios sino en conceptos y contenidos. Prácticamente el 100 % de la población a nivel europeo o americano ha jugado a los videojuegos y esta industria da empleo a miles de personas en todo el mundo.

1.5. Objetivos

Se pretende construir un videojuego tipo RTS, en la que el usuario pueda jugar una partida contra otro jugador, ya sea el ordenador u otra persona. Uno de los objetivos principales que cumple esta aplicación es la capacidad de ser multiplataforma donde personas con distintos sistemas operativos puedan jugar sin ser conscientes de la diferencia de plataforma.

Se ha desarrollado una aplicación funcionalmente completa, el jugador será capaz los parámetros del sistema a su gusto (resolución, aceleración, sonidos, subtítulos). Podrá jugar partidas de tipo refriega contra la máquina y también tiene la posibilidad de crear una partida para jugar a través de internet, ya sea por el uso de LAN o TCP/IP o unirse a una partida ya creada por otra persona siguiendo los protocolos anteriormente mencionados.

Los jugadores tendrán la posibilidad de elegir entre 3 bandos totalmente diferentes. Los Mercenarios, los Olvidados y el Cónclave cada una de ellas con gran variedad de unidades y edificios que ofrecerán una jugabilidad y capacidad de elaborar estrategias mayor para los jugadores.

Por último los jugadores tendrán la posibilidad de comunicarse a través de un chat proporcionado por el propio videojuego, tanto en la sala de espera como dentro de la partida.

Resumiendo los párrafos anteriores, los objetivos principales de este proyecto son los siguientes:

1. Crear un videojuego de tipo RTS 100 % funcional

2. Crear un videojuego multiplataforma
3. Capacidad de personalización de los aspectos técnicos del videojuego
4. Capacidad de jugar solo para entrenar y aprender
5. Capacidad de jugar contra otras personas a través de los protocolos LAN o TCP/IP
6. Capacidad de comunicación a través de un sistema de chat

1.6. Alcance

La creación de esta aplicación está orientada principalmente al entretenimiento de los usuarios de ordenadores personales donde podrán divertirse entrenando solos o contra un amigo.

El videojuego en su versión inicial solo dispondrá de 2 mapas disponible para jugar, eso sí en las dos modalidades existentes, un jugador y multijugador.

Se facilitarán manuales para que el usuario pueda crear los mapas a su antojo con los recursos disponibles y al ser código abierto el propio usuario si lo desea podrá modificar el comportamiento del sistema a todos los niveles, unidades, mapas, sistema de chat, entre otras, prologando así la vida del videojuego de manera casi permanente.

Capítulo 2

Juegos de Estrategia en Tiempo Real

En contra de la creencia popular, los juegos de estrategia en tiempo real no aparecieron con *Dune II: Battle for Arrakis* en 1992, ya que hay juegos como *The Ancient Art of War, Stonkers* y *Herzog Zwei*. No obstante este juego estableció un formato de juego revolucionario, tanto que muchos años después se siguió usando dicho formato para la realización de videojuegos de estrategia en tiempo real. Incluso hoy en día muchos juegos que llegan a ser top 1 en ventas a nivel mundial de este género tienen rastros del formato de aquel juego de tantos años atrás.

Fue el primer videojuego de este tipo que usaba el ratón para mover unidades, permitiendo a los jugadores interactuar de una forma fluida con sus tropas. [10]

Figura 2.1: Dune2: videojuego que marcó un hito en los RTS

2.1. Características de los RTS

Existen 3 características que cumplen la mayoría de los RTS:

- Recolección de recursos: Los jugadores deben recolectar recursos para ser capaces de crear unidades con las que jugar así como construir su base. Estos recursos varían dependiendo de la configuración del juego, pero por ejemplo los recursos pueden ser madera para construir casa o piedra para construir murallas fortificadas en la ciudad. [15]
- Desarrollo de la base: En la mayoría de los juegos RTS un jugador debe construir su base. Esta base puede ser cualquier cosa, desde una ciudad hasta un castillo pasando por una colonia espcaial. Esta base es el cuartel general del jugador y si es destruido o capturado puede causa rque el jugador pierda la partida. La base normalmente puede mejorarse usando los recursos recolectados por las unidades de dicho jugador. [15]
- Control directo de unidades: El jugador puede controlar las unidades móviles. Estas unidades pueden ser cualquier cosa desde tropas armadas, hasta extraterrestres de otra dimensión, pasando por magos y elfos dependiendo del tipo de RTS que se esté jugando. Los jugadores pueden dar órdenes a estas unidades (luchar, montar en barco y cruzar el océano, patrullar, etc) y hacerlo de forma individual o en grupo. Por ejemplo, un jugador podría necesitar mover un soldado o mover un grupo de 10 soldados y atacar con ellos como si se tratara de una sola unidad. [15]

2.2. Estado del Arte

A día de hoy existen muchos juegos tipo RTS en el mercado y aunque todos puedan parecer igual, todos tienen diseños muy característicos.

Lo primero que podemos diferenciar entre ellos es la ambientación que sus creadores les han dado. Ahí es muchas veces donde se realiza un mayor esfuerzo ya que todos los creadores de videojuegos quieren hacer que el suyo destaque por encima de los demás pero tal vez no sea la característica más relevante.

Otro punto que puede diferenciar completamente a un videojuego tipo RTS del resto es el enfoque que se le dé al control y manejo de las unidades. Con esto nos referimos a que existen dos formas de enfocar esto punto dentro de este tipo de videojuegos y aunque generalmente se dan las dos a la vez, una tiene más importancia que la otra. Nos referimos a macromanejo y micromanejo.

Entendemos por micromanejo como una serie de pequeños y detallados elementos del juego que deben ser controlados manualmente por el jugador. Este micromanejo puede ir orientado al combate o a la economía.

Cuando va orientado al combate lo que se busca es maximizar el daño ocasionado a las unidades móviles enemigas minimizando los daños en las unidades móviles del jugador. Las técnicas más comunes de micromanejo de unidades de combate por ejemplo son agrupar en formaciones las distintas unidades manteniendo, por ejemplo, las débiles unidades tipo arquero por detrás y protegidas por las fuertes unidades de cuerpo a cuerpo. Otro ejemplo es concentrar el

ataque de las unidades a distancia en las unidades enemigas para eliminarlas rápidamente o hacer uso de las relaciones *piedra-papel-tijeras* entre las distintas unidades, construyendo unidades que sirven específicamente para destruir las unidades que ha preparado el contrincante.

Cuando orientamos el micromanejo a la economía podemos estar hablando de cosas muy distintas ya que la economía en los RTS puede funcionar de diferentes maneras, pero principalmente consistirá en que las unidades tipo "trabajador o recolector" recolecten recursos o construyan edificios siempre y no haya ninguna ociosa. También comprende activar los sistemas de defensa para proteger a los trabajadores o darles órdenes para que escapen ante un ataque enemigo.

Un juego que es un ejemplo de un gran micromanejo de unidades es *Warcraft III: The frozen Throne*, ya que es un juego en el que poseeremos pocas unidades (30-35 como máximo) y cada unidad perdida es crucial para el juego.

Figura 2.2: Warcraft III: The Frozen Throne

Entendemos por macromanejo como el aspecto general de la economía del juego. Esto incluye construir edificios, desarrollar investigaciones y producir unidades, aparte de otras cosas que conllevan un gasto de los recursos recolectados por el jugador.

Por otro lado un juego que se centra mayormente en la economía y no tanto en las unidades es el *Age of Empires II: Age of Kings* ya que podremos poseer hasta 200 unidades y es primordial tener una buena economía para vencer a tu enemigo.

Figura 2.3: Age of Empires II: Age of Kings

Hasta ahora hemos hablado de juegos que son puramente RTS, pero existe gran variedad de videojuegos que son mixtos, es decir, tiene un género principal y luego en segundo plano contienen aspectos típicos de un RTS. Por ejemplo, un caso concreto es el famoso *Dungeon Keeper* donde somos el dueño de una mazmorra llena de monstruos y tenemos que vencer a los monstruos de la mazmorra enemiga. En principio parece un RTS en toda regla aunque realmente la gracia del juego está en meterte dentro del cuerpo de un monstruo de la mazmorra para aniquilar a los demás monstruos o simplemente matar gallinas.

Figura 2.4: Dungeon Keeper

Capítulo 3

Desarrollo del calendario

Se ha desarrollado un calendario para la creación de *Laboon*, dividimos todo el proceso de desarrollo de *Laboon* en una serie de bloques. Se muestran 2 calendarios, uno realizado antes del comienzo del proyecto, realizado mediante técnicas de estimación. El segundo es el calendario realizado al final del proyecto con el tiempo real que se ha tardado en desarrollar este proyecto.

3.1. Calendario Inicial

1. Un primer bloque que no aparece, representaría la duración del aprendizaje de SDL, la longitud de este bloque se corresponde casi con la duración total del proyecto debido a que según se iban desarrollando distinto módulos se deben adquirir nuevos conocimientos para avanzar en la construcción del mismo.

2. Configuración Forja: Este bloque se corresponde con la preparación del espacio ofrecido por la forja para instalar este proyecto, con todas sus utilidades.

3. Memoria Laboon: La documentación se ha ido realizando prácticamente desde el principio de la realización del proyecto, cada vez que se desarrollaba algo nuevo y se hacían las pruebas pertinentes, se documentaba convenientemente en este documento.

4. Análisis Laboon: Este conjunto de tareas se corresponde con la realización de los casos de uso, el diagrama de clases básico de *Laboon* y los diagramas de comportamiento correspondientes a los distintos casos de uso.

5. Análisis Menú: Diagramas y descripciones de casos de uso de las funciones que debe cumplir la parte del menú de *Laboon*. Realización de los diagramas de comportamiento correspondientes.

6. Análisis Juego: Diagramas y descripciones de casos de uso de las funciones que debe cumplir la parte del juego de *Laboon*. Realización de los diagramas de comportamiento correspondientes.

7. Diseño Laboon: Realización de los diagramas de secuencia y la elección de la arquitectura de *Laboon*.

8. Diseño Menú: Realización de los diagramas de secuencia correspondientes a la parte del menú de *Laboon*.

9. Diseño Juego: Realización de los diagramas de secuencia correspondientes a la parte del juego de *Laboon*.

10. Implementación: Codificación de todas las funciones de *Laboon*.

11. Botones MenuInicio: Realización del diseño general de los botones del menú, construcción de cada botón y posterior codificación en *Laboon*. Construcción de un sistema de captura de eventos para todos los botones del menú de *Laboon*.

12. Submenú opciones: Codificación de las distintas opciones del submenú de opciones (Resolución, Aceleración, Sonidos, Subtítulos).

13. Ficheros Configuración: Preparación de los ficheros de configuración de video y de audio de *Laboon*.

14. Submenú Elección Bando: Diseño y codificación del submenú de elección de bando del jugador.

15. Submenú UnJugador: Diseño y codificación del submenú para un jugador, incluyendo selección de bando y mapa.

16. Menú Multijugador: Diseño y codificación del menú multijugador incluyendo todas las pantallas necesarias para comenzar una partida.

17. Conexiones Base: Codificación y diseño de la arquitectura P2P multijugador, uso de bibliotecas conexión a través de red.

18. Sistema Chat: Diseño y codificación de un sistema de chat basado en conexiones base.

19. Juego: Codificación de todas las entidades referentes a la parte del juego de *Laboon* con el correspondiente diseño gráfico de sus elementos.

20. Tablero: Codificación del tablero del juego y de un sistema de construcción propios de tableros para el jugador, establecimiento de las dimensiones del tablero y captura de eventos en el mismo.

21. Diseño Unidades: Elección del diseño gráfico de todas las unidades de *Laboon*.

22. Diseño Edificios: Elección del diseño gráfico de todas los edificios de *Laboon*.

23. Punteros: Diseño gráfico de los punteros de ratón para *Laboon*.

24. MiniMapa: Diseño gráfico y construcción de un minimapa usando técnicas de reducción de imágenes. El método bicúbico se ha usado.

25. Acciones Unidades: Codificación de todas las acciones básicas y especiales de cada unidad de *Laboon*.

26. Acciones Edificios: Codificación de todas las acciones básicas y especiales de cada edificio de *Laboon*.

27. Pruebas y Modificaciones: Realización de las pruebas unitarias de cada módulo de *Laboon* y algunas pruebas funcionales en modo un jugador y multijugador. Tras la realización de las pruebas se modificaron los módulos con errores de ejecución no detectados.

28. Manual Usuario: Construcción y diseño del manual de usuario, donde se dan a conocer todas las funciones del juego y de las unidades de cada bando.

29. Unificación Código: Proceso de unificación de código entre Windows y Linux para reducir al mínimo las diferencias entre uno y otro para futuras modificaciones.

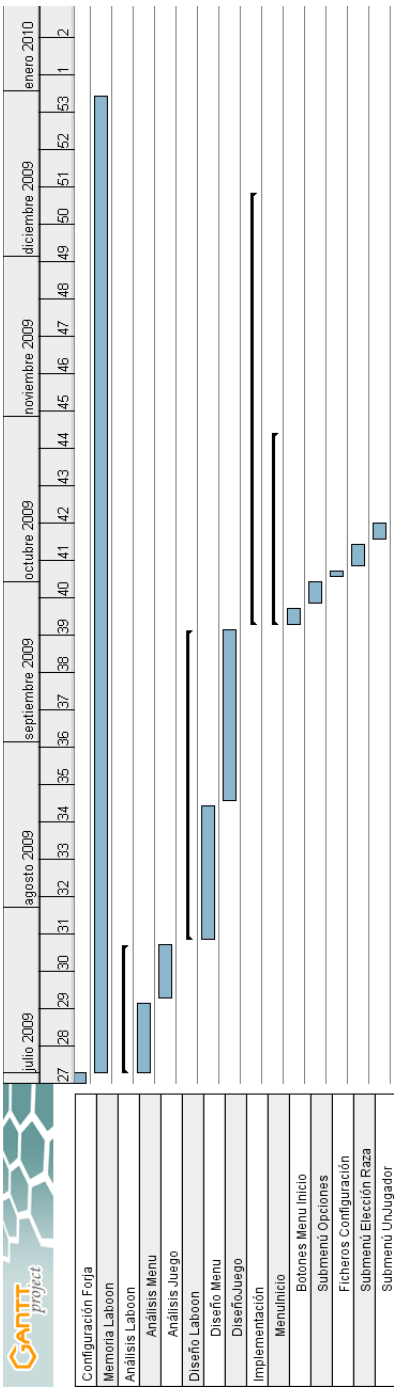

Figura 3.1: Calendario Planificado (Parte1)

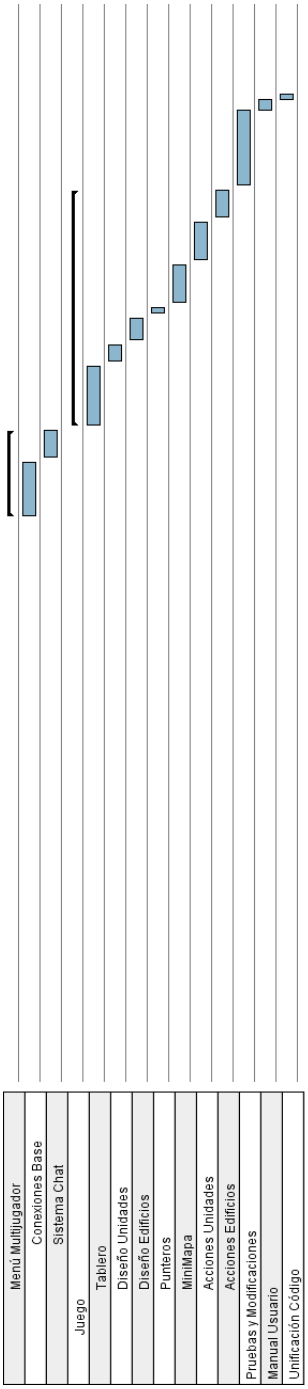

Figura 3.2: Calendario Planificado (Parte2)

3.2. Calendario Final

Aunque las tareas en ambos calendarios son las mismas ya que se ha intentado seguir lo máximo posible la planificación inicial, la duración de las tareas ha variado. Se ha producido una variación considerable en algunas tareas, siendo la duración real mayor que la duración estimada inicialmente. Estos errores de estimación se producen principalmente por la falta de conocimiento inicial de esta plataforma así como por la falta de experiencia realizando estimaciones.

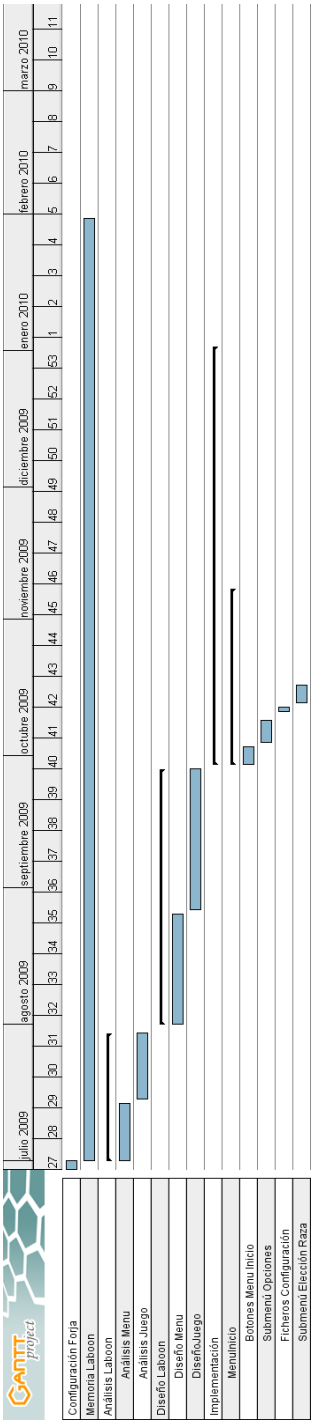

Figura 3.3: Calendario Real (Parte1)

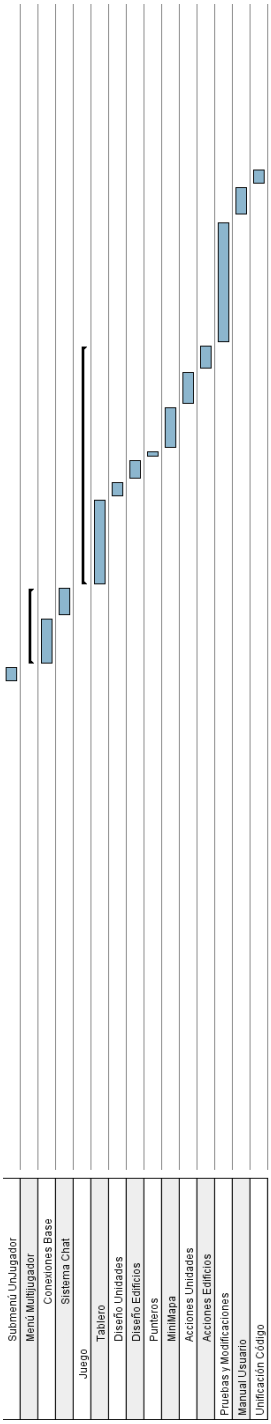

Figura 3.4: Calendario Real (Parte2)

Capítulo 4

Descripción General de Laboon

Laboon ofrecerá la posibilidad de que el usuario pueda jugar a un juego tipo RTS ya sea contra el propio ordenador o contra otra persona a través de internet. Este producto se puede considerar como un producto completo e independiente ya que no forma parte de ningún sistema mayor.

Aparte existe la posibilidad de configurar varios aspectos en relación a las características del producto, las cuales se detallarán más adelante.

4.1. Interfaces

Este producto no posee ninguna interfaz externa ya que se puede considerar una versión *Standalone* que se instalará en la máquina en la que se vaya a usar.

4.1.1. Interfaces de Usuario

Aunque la interfaz de usuario en este caso es única hay dos partes de esta aplicación bien diferenciadas.

- Menú: En esta parte de la aplicación podremos elegir qué tipo de partida queremos jugar, en qué mapa jugar y con qué bando jugar. En la sección de opciones podremos modificar las siguientes características del sistema: Resolución, Sonidos, Aceleración y Subtítulos. Estas características serán comentadas a pleno detalle más adelante.

Figura 4.1: Imagen del menú principal de Laboon

- Juego: Esta parte de la aplicación es mucho más compleja ya que es la disposición y cantidad de elementos es variable dependiendo de las acciones que realicemos: Marcar una unidad propia, marcar un grupo de unidades propias, marcar una unidad enemiga, marcar un edificio propio... La pauta que siempre se cumplirá es una visión de la zona del mapa en la que nos encontremos y un minimapa que nos mostrará el estado actual del campo de batalla completo.

Figura 4.2: Imagen de Laboon

4.1.2. Interfaces con el Hardware

No existen unos requisitos específicos de hardware para el correcto funcionamiento de *Laboon* ya que está construido usando unas bibliotecas con capacidad de funcionar en varias plataformas. El único requisito que se podría nombrar dentro de este grupo es que todos los archivos fuentes y de recursos se encuentren donde lo indique el manual de instalación.

4.1.3. Características de los usuarios

Laboon va dirigido a usuarios finales de escritorio y no precisa de ningún tipo de formación previa para su uso. Solo se exigen los conocimientos mínimos sobre el manejo de los ordenadores y seguir con precisión las instrucciones de instalación y de uso.

4.2. Restricciones Generales

Laboon ha sido creado teniendo en cuenta una serie de restricciones.

La máquina donde se vaya a proceder a ejecutar deberá tener todas las bibliotecas necesarias para la correcta ejecución del videojuego, cuyo proceso de instalación viene detallado en el manual de instalación. En concreto: [10]

- SDL (Simple DirectMedia Layer). Conjunto de bibliotecas que proporcionan funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. [10]
- SDL_Image. Extiende notablemente las capacidades para trabajar con diferentes formatos de imagen.
- SDL_Net. Proporciona funciones y tipos de dato multiplataforma para programar aplicaciones que trabajen con redes.
- SDL_Mixer. Extiende las capacidades de SDL para la gestión y uso de sonido y música en aplicaciones y juegos.
- SDL_TTF. Permite usar fuentes *TrueType* en aplicaciones SDL.

Respecto al menú de inicio de *Laboon* el usuario solo podrá o jugar contra el ordenador en la opción **Un Jugador** o tendrá que jugar contra una persona en la opción **Multijugador**.

El usuario solo podrá modificar las características técnicas que le ofrece dicho menú, que son las que siguen:

- Resolución: El usuario podrá elegir entre 3 posibles resoluciones: 800x600, 1024x768, 1280x1024.
- Aceleración: El usuario podrá activar o desactivar la aceleración Hardware.
- Sonidos: El usuario podrá activar o desactivar el sonido del videojuego.

- Subtítulos: El usuario podrá activar o desactivar los subtítulos que aparezcan en el videojuego.

Cuando el usuario se disponga a jugar una partida solo podrá elegir un mapa donde jugar, esto se cumple tanto para **Un jugador** como para **Multijugador**, a su vez solo podrá elegir 1 bando entre los 3 disponibles del juego, cada uno de ellos con bonificaciones únicas:

- Cónclave.
- Mercenarios.
- Olvidados.

En cuanto a los requisitos técnicos del sistema, *Laboon* solo necesita que el usuario posea una tarjeta gráfica, tarjeta de sonido y posibilidad de conexión a Internet o conexión a una red local si el usuario quiere entablar una partida **Multijugador**.

4.3. Restricciones de Laboon

Laboon ha sido diseñado y construido para satisfacer todos los requisitos mínimos de un juego tipo *RTS*, pero también está construido bajo una serie de restricciones.

- Partidas:
  - Sólo existen 2 tipos de partidas en *Laboon*. **Un Jugador** y **Multijugador**, independiente del tipo de partida solo se permitirán 2 participantes en una partida.
- Mapas:
  - Los mapas de *Laboon* deberán tener dimensiones de 64 x 64 a 128 x 128 casillas de dimensión. No tiene porqué ser cuadrado.
  - Los recursos en los mapas son limitados.
- Unidades:
  - Todas las unidades de *Laboon* son ortogonales, por tanto solo existen 8 direcciones posibles a la hora de moverse.
  - Las unidades serán acuáticas o terrestres, no existiendo unidades aéreas.
  - Cada jugador tiene una población máxima limitada de 100.
  - Todos los vehículos/edificios/tecnologías tienen un coste asociado en **Metal**, **Madera** y **Población**.
  - Las unidades solo atacarán a objetivos enemigos si éstos están dentro de su rango de acción, en caso contrario se acercarán.
  - Será imposible construir una unidad si no posees suficientes recursos y/o población disponible.

- Se podrán construir las unidades de de manera individual, no pudiendo establecer una cola de creación de unidades.
- La unidad **TanqueTransporte** carga las unidades de manera individual y las libera todas a la vez.
- La unidad **TanqueTransporte** solo puede cargar unidades pequeñas, con una capacidad máxima de 5 unidades.
- Sólo se podrán construir edificios en las zonas habilitadas para ello siempre y cuando no estén ocupadas por otros edificios/unidades.
- Sólo se pueden recolectar recursos en las zonas habilitadas para ello.
- Las tareas de recolección de recursos, construcción y reparación sólo lo pueden realizar unidades especializadas en cada caso.
- Los índices de comercio entre **Metal** y **Madera** son variables según la demanda de cada recurso y dichos índices son compartidos por ambos contendientes.

4.4. Funcionalidades adicionales de Laboon

Se han implementado varias funcionalidades adicionales en *Laboon* para que la curva de aprendizaje de la aplicación sea menor así como para proporcionar una mayor agilidad al juego cuando el usuario haya dominado las funciones básicas.

Principalmente la funcionalidad que hace que la curva de aprendizaje sea menor, ha sido el uso de *tooltips*.

Un tooltip es un elemento de las interfaces gráficas que se emplea junto con el cursor del ratón. Cuando un usuario posiciona su cursor sobre un elemento (sin hacer click sobre este) un pequeño rectángulo aparece para brindar información relacionada a ese elemento.

Con esto podemos conseguir que el usuario evite tener que leerse el manual del jugador completo para poder empezar a jugar a *Laboon*.

Enfocándonos en jugadores más expertos podemos decir que *Laboon* proporciona 2 funcionalidades que no todos los juegos tipo *RTS* proporcionan:

- Personalización de grupos: *Laboon* posee la capacidad de crear 9 grupos personalizables de unidades, donde el usuario tiene que pulsar la tecla Control + X (donde $X \in [1, 9]$) para crear los grupos y luego pulsarán dicho número para seleccionar las unidades de cada grupo.
- Teclas de acceso directo: Casi todas las funcionalidades existentes en el juego se podrán hacer usando el teclado del ordenador, para dar mayor velocidad a las acciones del usuario. Todas estas acciones vienen definidas al final del manual de usuario contemplado en la sección 10.2.

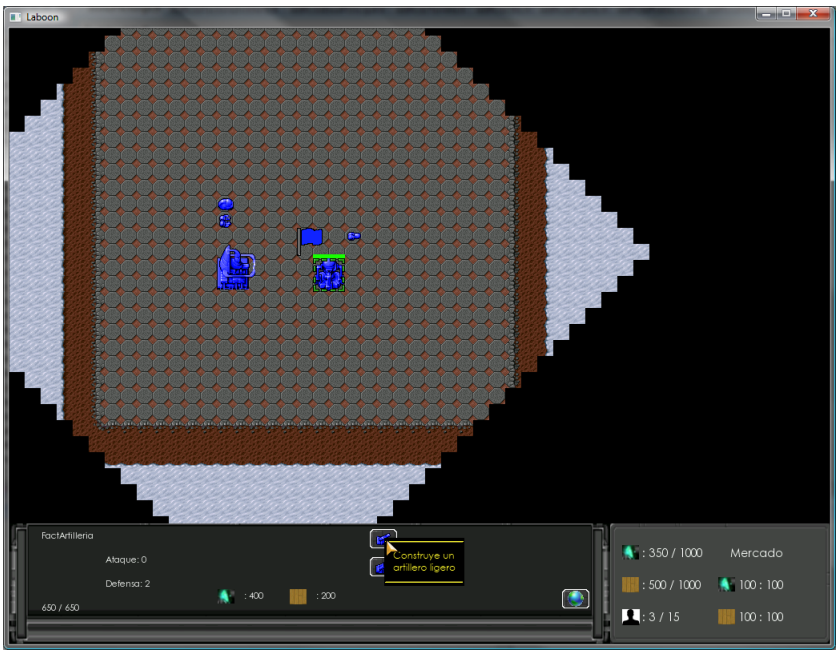

Figura 4.3: Ejemplo de Tooltip

4.5. Requisitos para futuras versiones

Para futuras versiones existen ciertos aspectos para mejorar *Laboon*.

Todo el entorno gráfico se podría mejorar notablemente con la ayuda de algún diseñador o grafista que se encargara de todos los aspectos visuales del videojuego.

Respecto a los bandos del videojuego se aumentarían las diferencias para personalizarlas más. Otro aspecto a mejorar sería aumentar la cantidad de jugadores que pudieran jugar en una partida, tanto a nivel de **Un Jugador** como en **Multijugador**.

Otro aspecto dentro del videojuego sería mejorar el *Lore* del mismo, es decir crear una historia con un trasfondo para añadirle al videojuego un poco de consistencia.

Por último se podría crear una aplicación paralela al videojuego, un *Map Editor*, para facilitar al usuario crear los mapas del juego y poder imponer sus propias reglas al mismo. Con esto se conseguiría que el usuario creara sus propias fases y campañas.

4.6. Elección de Herramientas

4.6.1. Constructor de Diagramas

Se han estudiado varias herramientas en distintos ámbitos. Por ejemplo a la hora de realizar los diagramas; existen muchos programas muy buenos para realizar todos los diagramas necesarios para lo referente a este documento, uno de los más conocidos es el **Dia**, pero finalmente nos hemos decidido por usar **MagicDraw** ya que aunque los dos programas tienen una funcionalidad similar, a nivel de visualización **MagicDraw** tiene una mejor presentación.

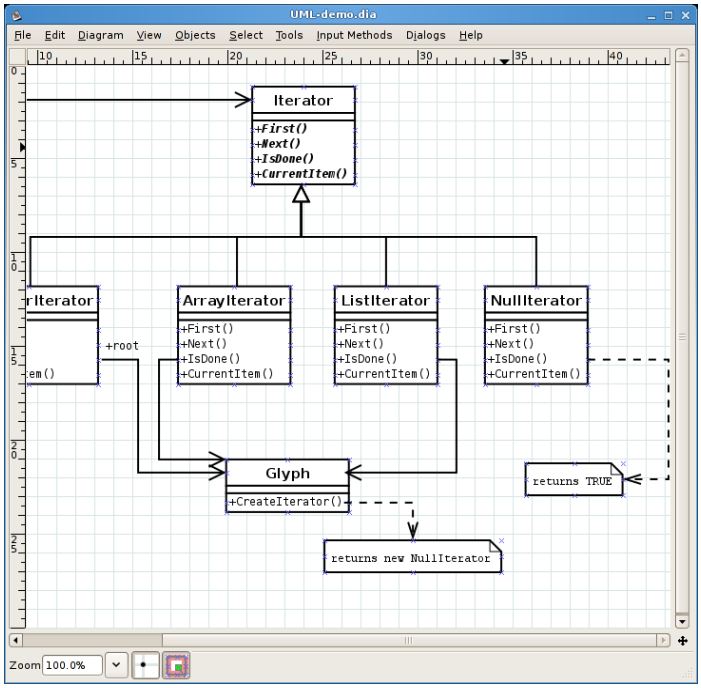

Figura 4.4: Constructor de diagramas: DIA

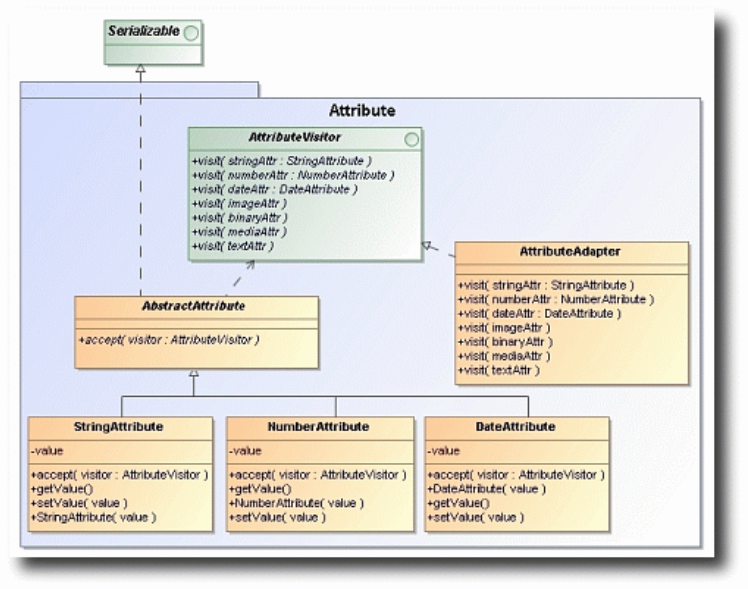

Figura 4.5: Constructor de diagramas: MagicDraw

4.6.2. Documentación

A la hora de realizar la documentación nos hemos inclinado por la aplicación **LaTeX** frente a otras con **OpenOffice** o **Office 2003/07** debido a la gran capacidad que tiene el compilador de **LaTeX** para construir textos con símbolos complejos, como por ejemplo textos matemáticos. Por otra parte trae muchas automatizaciones que ayudan al usuario a construir textos complejos sin mucha dificultad.

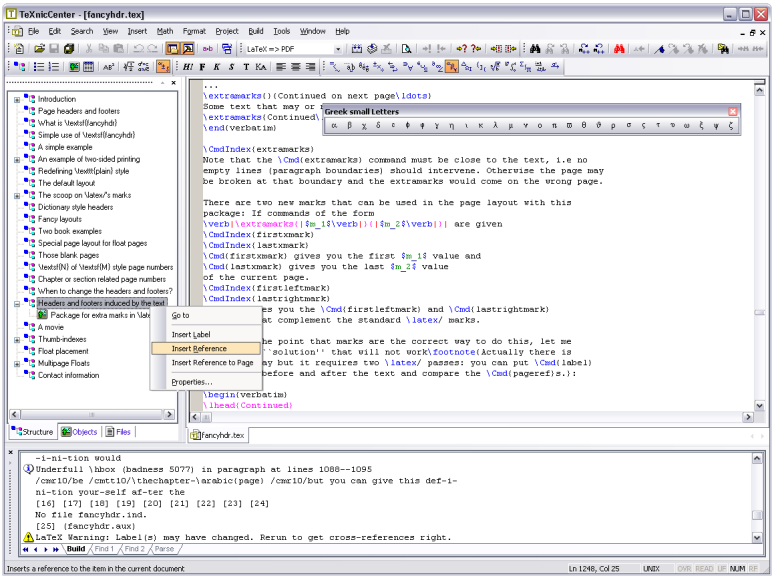

Figura 4.6: Compilador LaTeX con entorno TexnicCenter

4.6.3. Bibliotecas Gráficas

Existen gran variedad de bibliotecas gráficas para la construcción de videojuegos en innumerables plataformas. Para la realización de este proyecto se tuvieron en cuenta 3 conjuntos de bibliotecas para desarrollar *Laboon*.

- XNA
- Ogre3D
- libSDL

XNA

Microsoft XNA es un conjunto de herramientas con un entorno de ejecución administrado proporcionado por Microsoft que facilita el desarrollo de juegos de ordenador y de gestión. Intentos para liberar a los desarrolladores de juegos la creación de "repetitivo código" y traer diferentes aspectos de la producción de juego en conjunto de un único sistema. El XNA es una herramienta que se anunció el 24 de marzo de 2004, en la Game Developers Conference en San José, California. La primera comunidad Technology Preview de XNA Build fue lanzada el 14 de marzo de 2006. XNA Game Studio 2.0 fue lanzado en diciembre de 2007, seguida de XNA Game Studio 3.0 en 30 de octubre de 2008. [10]

XNA actualmente abarca secciones de Microsoft Game Development Sections, incluyendo el estándar Kit de desarrollo de Xbox y XNA Game Studio. [10]

XNA Framework

XNA Framework se basa en la implementación nativa de .NET Compact Framework 2.0 para el desarrollo de la Xbox 360 y .NET Framework 2.0 en Windows. Incluye un amplio conjunto de bibliotecas de clases, específicos para el desarrollo de juegos, para promover la reutilización de código máximo a través de plataformas de destino. El marco se ejecuta en una versión de Common Language Runtime que se ha optimizado para que los juegos de azar proporcionar un entorno de ejecución administrado. El tiempo de ejecución está disponible para Windows XP, Windows Vista y Xbox 360. Dado que XNA Games están programados para el tiempo de ejecución, que se ejecuten en cualquier plataforma que admite el XNA Framework con mínima o ninguna modificación. Juegos que se ejecutan en el marco técnicamente pueden escribirse en cualquier lenguaje compatible con .NET, pero oficialmente se admiten sólo C# y XNA Game Studio Express IDE y todas las versiones de Visual Studio 2005. [10]

El XNA Framework por lo tanto encapsulado bajo nivel tecnológicos detalles relacionados en un juego de codificación, asegurándose de que el propio marco se encarga de la diferencia entre plataformas cuando juegos son portados desde una plataforma compatible a otro y lo que permite que los desarrolladores de juegos para concentrarse más en el contenido y la experiencia de juego. XNA Framework se integra con una serie de herramientas, tales como la plataforma múltiple audio creación herramienta (XACT), para ayudar en la creación de contenido. XNA Framework proporciona apoyo para la creación de juego 2D y 3D y permite el uso de los controladores de Xbox 360 y vibraciones. Sólo los juegos de marco XNA actualmente destinadas a la plataforma Xbox se pueden

ser distribuidos a los miembros de Club de la Microsoft XNA Creator que lleva un 99 dólares/año suscripción.3 Desktop application pueden ser distribuidas gratuitamente bajo licencia actual de Microsoft. [10]

XNA Build

XNA Build es un conjunto de herramientas de gestión de canalización de activos de juego, que por definición, mantener, depuración y optimización de la canalización de juego activo de los esfuerzos de desarrollo de juegos individuales. Una tubería de juego activo describe el proceso de por qué contenido de juegos, tales como texturas y modelos 3D, se modifican para un formato adecuado para su uso por el motor de juegos de azar. XNA cree ayuda a identificar las dependencias de la tubería y también proporciona acceso de la API para habilitar el tratamiento ulterior de los datos de dependencia. Para ayudar a reducir el tamaño de un juego por encontrar contenido que no se utiliza en realidad, se pueden analizar los datos de dependencia. Por ejemplo, generar XNA análisis reveló que el 40 % de las texturas que acompaña MechCommander 2 eran no utilizado y podría se han omitido. [10]

XNA Game Studio

XNA Game Studio es un entorno de desarrollo integrado (IDE) para el desarrollo de juegos. Cinco revisiones han sido liberados hasta ahora. [10]

Ogre3D

OGRE 3D (acrónimo del inglés Object-Oriented Graphics Rendering Engine) es un motor de renderizado 3D orientado a escenas, escrito en el lenguaje de programación C++. [10]

Sus bibliotecas evitan la dificultad de la utilización de capas inferiores de bibliotecas gráficas como OpenGL y Direct3D, y además, proveen una interfaz basada en objetos del mundo y otras clases de alto nivel. [10]

El motor es software libre, licenciado bajo LGPL y con una comunidad muy activa. Incluso ha sido utilizado en algunos videojuegos comerciales, como por ejemplo *Ankh* y *Earth Eternal*. [10]

libSDL

Simple DirectMedia Layer (SDL) es un conjunto de bibliotecas desarrolladas con el lenguaje C que proporcionan funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. [10]

Pese a estar programado en C, tiene wrappers a otros lenguajes de programación como C++, Ada, C#, Basic, Erlang, Lua, Java, Python, etc. También proporciona herramientas para el desarrollo de videojuegos y aplicaciones multimedia. Una de sus grandes virtudes es el tratarse de una biblioteca multiplataforma, soportando oficialmente los sistemas windows, GNU/Linux, MacOS y QNX, además de otras arquitecturas/sistemas como Dreamcast, GP32, GP2X... De ahí le vienen las siglas Simple Directmedia Layer que más o menos alude a capa de abstracción multimedia. Desarrolladas inicialmente por Sam Lantinga, desarrollador de videojuegos para la plataforma GNU/linux. [10]

La biblioteca se distribuye bajo la licencia LGPL, que es la que ha provocado el gran avance y evolución de las SDL. [10]

Biblioteca elegida

Finalmente entre estas opciones elegimos libSDL frente a las otras dos. Las razones principales son que finalmente se decidió construir un juego bajo una plataforma 2D, lo cual hacía que Ogre3D perdiera gran parte de su capacidad y fue descartado. Otro objetivo que debía cumplir la biblioteca que fuéramos a escoger que fuera capaz de funcionar tanto en sistemas Windows como en sistemas linux y a ser posible bajo licencia GPL, así que se descartó XNA por esos motivos aún siendo probablemente un motor mucho más potente que libSDL.

4.7. Herramientas utilizadas

Para la realización de este proyecto se han usado herramientas de software libre en su mayoría, junto con otras herramientas gratuitas, minimizando de esta manera en costes, usando herramientas fiables, seguras y estables.

Software libre utilizado para la creación de la aplicación:

- Sistema Operativo Libre: Ubuntu 9.04 Jaunty Jackalope. Sistema operativo libre con licencia GNU.
- IDE de desarrollo: Eclipse usando C++.
- Bibliotecas LibSDL. Bibliotecas con licencia GNU.
- Modelado de figuras: Blender. Software con licencia GNU.
- Pintado 2D: Gimp. Software con licencia GNU.

Software adicional usado para la construcción de esta aplicación. El objetivo de usar estas aplicaciones es garantizar una compatibilidad de uso con los sistemas Windows, ya que son los más usados.

- Sistema operativo: Windows Vista Home Premium.
- IDE de desarrollo: Visual Studio Express 2008 usando C++.
- Pintado 2D: Aplicación de dibujado: Paint.Net.

Software utilizado para realizar la documentación:

- Editor de documentación: Latex.
- Compilador de documentación: TeXnicCenter.
- Editor de Diagramas: DrawMagic v16.6.

4.8. Comunidad

Desde el inicio del desarrollo de *Laboon* se han ido incorporando poco a poco diferentes personas a la causa para ayudarme de forma desinteresada a desarrollar este proyecto.

Integrantes de la comunidad Laboon:

- José Cordero: Creador de *Laboon*.
- Manuel Palomo: Director del proyecto. Manolo me ha ayudado con varios aspectos técnicos del videojuego.
- Álvaro Gil: Tester. Álvaro me ha ayudado a probar muchos aspectos del videojuego empleando muchas horas.
- José Sánchez: Creador de la OST. José ha creado varias pistas de música para ambientar el videojuego.
- Jesús Cordero: Diseñador. Jesús ha diseñado el interfaz gráfica del menú inicial del videojuego.
- Lorena Gutiérrez: Supervisora. Lorena ha desarrollado la web donde está colgado *Laboon* y me ha ayudado a modificar varios aspectos técnicos del videojuego.

A todos ellos, muchas gracias.

Capítulo 5

Desarrollo del Proyecto

Durante los siguientes capítulos analizaremos los requisitos del sistema aplicando una metodología de desarrollo.

5.1. Metodología de Desarrollo

Para el desarrollo y especificación de *Laboon* utilizaremos como metodología el *RUP* (Rational Unified Process). Mediante esta metodología desarrollaremos la aplicación orientada a objetos. Utilizaremos *UML* como lenguaje de modelado y notación.

La principal razón por la que se ha desarrollado la aplicación mediante una metodología orientada a objetos y notación UML, es la escalabilidad, el poder añadir en un futuro nuevos módulos sin tener que modificar el diseño existente. Se han aprovechado otras características singulares de este paradigma tales como la herencia, mayor legibilidad del código, mayor reusabilidad y mantenimiento.

5.2. Especificación de requisitos del sistema

5.2.1. Requisitos de interfaces

En la sección del menú del videojuego la interfaz con el sistema se hace a través de dos clases. Estas clases únicamente acceden a una serie de ficheros de configuración a partir de los cuales se aplicará la configuración deseada al videojuego.

- ConfiguracionVideo: Clase que se encarga de obtener información del fichero *video.cfg*. De este fichero podemos obtener: La resolución de la aplicación, el tipo de aceleración, aparición de subtítulos durante el videojuego.
- ConfiguracionSonido: Clase que se ocupa de obtener información del fichero *audio.cfg*. De este fichero podemos obtener la ubicación de los archivos de sonido, el volumen al que se reproducirán los archivos de audio y la cantidad de archivos que componen la banda sonora personalizable del videojuego.

En la sección del juego en sí no clases que interaccionen con el sistema operativo, excluyendo los ficheros de imagen que representan a dichas unidades. En todo momento se trabaja a través de la capa creada por las bibliotecas LibSDL.

Para las interfaces de usuario hemos usado las bibliotecas LibSDL v1.2 y se ha construido según viene documentado en la sección 4.1.1

Parte II

Análisis y Diseño de Laboon

Capítulo 6

Análisis del Sistema

6.1. Modelos de casos de uso

Para el desarrollo de un proyecto es muy importante realizar un modelado tanto de análisis y diseño para obtener un conjunto de modelos que nos permitan especificar los requisitos, la estructura y el comportamiento del sistema.

Comenzaremos nuestro análisis del sistema realizando el modelado de casos de uso. Entendemos un caso de uso como una descripción que especifica un comportamiento deseado del sistema. Los casos de uso describen qué hace el sistema representando los requisitos funcionales.

Para describir los casos de uso de *Laboon*, nos ayudaremos de representaciones gráficas, diagramas de casos de uso, y de una descripción del mismo. Ésta descripción incluirá el nombre del caso de uso, actores que intervienen en él, precondiciones, postcondiciones, escenario principal y lo posibles escenarios alternativos. Entendemos por actores del sistema aquellos usuarios, sistemas o tiempo que intervienen en la aplicación. Entendemos por escenario principal, el flujo normal que seguirá la aplicación para ese caso de uso. Entendemos por escenarios alternativos aquellas variantes que se pueden dar durante el transcurso del caso de uso(errores, otras opciones, etc).

6.1.1. Diagramas de Casos de Uso

Comenzaremos realizando un agrupamiento de casos de uso. Posteriormente iremos descendiendo en abstracción, hasta los casos de usos más concretos. En un nivel mayor nos encontramos con dos casos de uso. El primero de ellos es **Gestión Menú** que se ocupa de todas las acciones que se pueden realizar cuando nos encontramos en el menú de *Laboon*, y el segundo es **Gestión Videojuego** que análogamente se encargará de todas las acciones a realizar cuando el videojuego en sí se encuentre en funcionamiento.

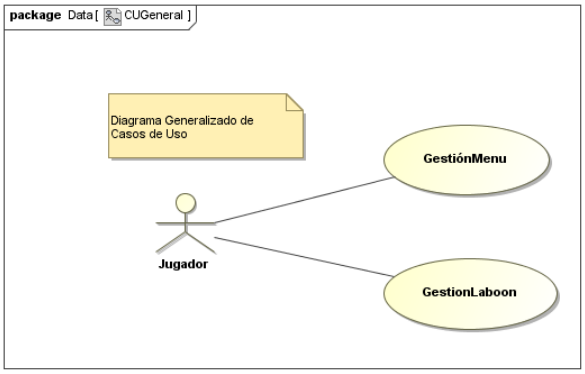

Figura 6.1: Diagrama de caso de uso general.

Dentro de *Gestión Menú* existen varios casos de uso agrupados según la opción que elijamos. La primera es jugar en modo *unJugador*, la segunda es jugar en modo *multiJugador*, la tercera es entrar en *opciones* y por último tendremos *salir*.

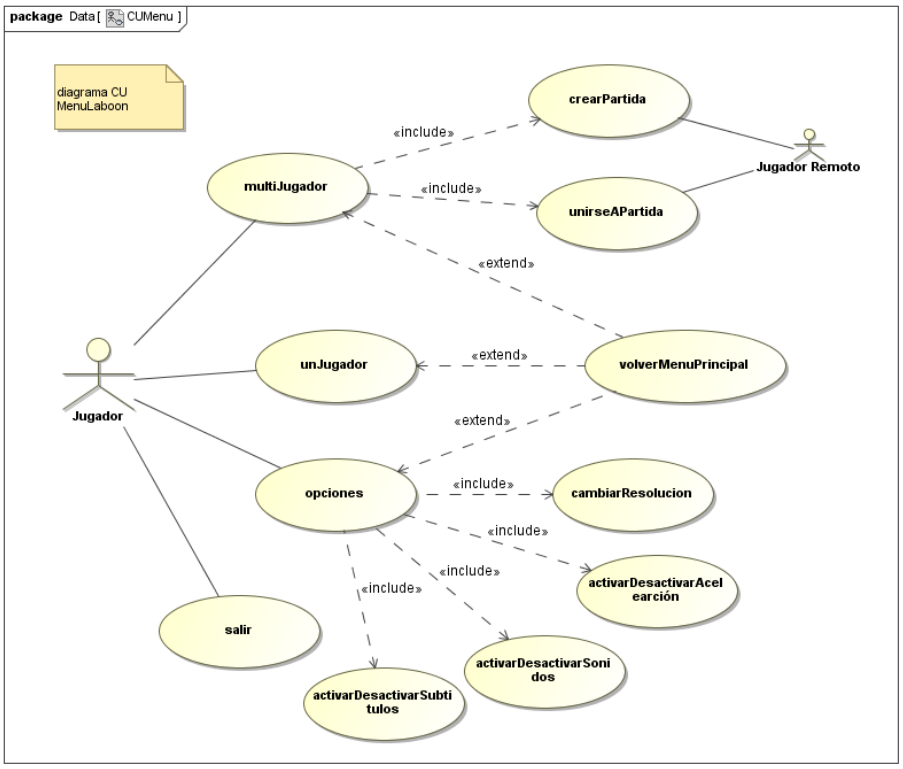

Figura 6.2: Diagrama de caso de uso del menú de *Laboon*.

6.1.2. Caso de Uso: Un Jugador

Caso de Uso: Un Jugador.

Precondiciones: Ninguna

Postcondiciones: El jugador establecerá una partida en modo "Un Jugador" contra el sistema.

Descripción: Partida de un solo jugador.

Resumen: El jugador quiere jugar una partida de refriega contra el sistema.

Actores: Jugador (Principal)

Escenario Principal

1. Jugador selecciona la opción "Un Jugador".
2. El sistema asigna el tipo de partida a *unJugador* y devuelve la pantalla de selección de mapas.
3. El jugador elige el mapa que desee jugar y le da a "Aceptar".
4. El sistema guarda el mapa seleccionado devuelve la pantalla de selección de bando.
5. El jugador selecciona la bando con el que desea jugar y pulsa "Aceptar".
6. El sistema guarda la bando seleccionado por el jugador y comienza la partida.

Escenarios Alternativos

- *a. El jugador cancela la petición de partida contra el sistema.

6.1.3. Caso de Uso: Crear Partida Multijugador

Caso de Uso: Crear Partida Multijugador.

Precondiciones: Ninguna

Postcondiciones: El jugador iniciará una partida en modo "Multijugador" contra el jugador remoto, siendo el primero el host de la partida.

Descripción: Ser Host para partida multijugador.

Resumen: El jugador participará como servidor en una partida multijugador.

Actores: Jugador (Principal) y Jugador Remoto (secundario)

Escenario Principal

1. El jugador selecciona la opción "Multijugador".
2. El sistema devuelve la pantalla de elección de tipo de partida multijugador.
3. El jugador selecciona la opción "Crear Partida".´

4. El sistema devuelve un formulario para que el jugador introduzca su nombre.
5. El jugador introduce su nombre, pulsa la tecla “Intro” y pulsa “Aceptar”.
6. El sistema guarda el nombre del jugador y devuelve la pantalla de selección de mapas.
7. El jugador elige el mapa que desea y selecciona “Aceptar”.
8. El sistema almacena el mapa seleccionado y muestra la pantalla de selección de bando.
9. El jugador selecciona la bando con el que desea jugar y selecciona “Aceptar”.
10. El sistema almacena la bando seleccionado por el jugador y muestra la pantalla de partida multijugador.
11. El jugador pulsa el botón “Esperar” para que el Jugador Remoto acceda a la partida.
12. El jugador remoto accede a la partida y pulsa el botón “Listo”.
13. El jugador pulsa el botón “Comenzar”.
14. El sistema lanza la partida multijugador.

Escenarios Alternativos

- *a. El jugador cancela el proceso de partida multijugador.

6.1.4. Caso de Uso: Unirse a una Partida Multijugador.

Caso de Uso: Unirse a una Partida Multijugador.

Precondiciones: Ninguna

Postcondiciones: El jugador participará en una partida tipo “Multijugador”, actuando como cliente en la misma.

Descripción: Ser cliente en una partida multijugador.

Resumen: El jugador participará como cliente en una partida multijugador.

Actores: Jugador (Principal) y Jugador Remoto (secundario).

Escenario Principal

1. El jugador selecciona la opción “Multijugador”.
2. El sistema devuelve la pantalla de elección de tipo de partida multijugador.
3. El jugador selecciona la opción “Unirse a Partida”.
4. El sistema devuelve el formulario para que el jugador introduzca su nombre.

5. El jugador introduce su nombre, pulsa la tecla “Intro” y pulsa el botón “Aceptar”.
6. El sistema guarda el nombre del jugador y devuelve la pantalla de selección de bando.
7. El jugador selecciona la bando con el que desea jugar y selecciona “Aceptar”.
8. El sistema devuelve el formulario para que el jugador introduzca la IP del jugador remoto.
9. El jugador introduce la dirección IP del jugador remoto , pulsa la tecla “Intro” y pulsa “Aceptar”.
10. El sistema comprueba la IP introducida y conecta con el jugador Remoto y muestra la pantalla de partida multijugador.
11. El jugador pulsa el botón “Listo”.
12. El sistema lanza la partida multijugador.

Escenarios Alternativos

- *a. El jugador cancela el proceso de partida multijugador.
- 10b. El sistema no puede conectar a la partida del jugador remoto.

6.1.5. Caso de Uso: Cambiar Resolución

Caso de Uso: Cambiar Resolución

Precondiciones: El fichero de configuración de video debe existir.

Postcondiciones: El jugador cambiará la propiedad *Resolución* del sistema.

Descripción: Cambiar la resolución del sistema.

Resumen: El jugador quiere modificar la resolución existente en el sistema por otra que le sea más cómoda.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón “Opciones”.
2. El sistema devuelve la pantalla con las distintas opciones disponibles.
3. El jugador selecciona el botón de “Resolución”.
4. El sistema muestra la pantalla con las resoluciones disponibles en el sistema.
5. El jugador selecciona la resolución deseada y pulsa el botón “Aceptar”.
6. El sistema guarda los cambios y actualiza la resolución actual por la deseada.

Escenarios Alternativos

- *a. El jugador cancela el proceso de cambio de resolución.

6.1.6. Caso de Uso: Activar o Desactivar Aceleración Hardware

Caso de Uso: Activar o Desactivar Aceleración Hardware

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones: El jugador cambiará la propiedad *Aceleración* del sistema.

Descripción: Cambiar la aceleración del sistema.

Resumen: El jugador quiere modificar la aceleración actual de Hardware a Software o viceversa.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Opciones".
2. El sistema devuelve la pantalla con las distintas opciones disponibles.
3. El jugador selecciona el botón "Aceleración".
4. El sistema muestra los distintos tipos de aceleraciones existentes.
5. El jugador elige la aceleración deseada y pulsa "Aceptar".
6. El sistema guarda los cambios y actualiza la aceleración del mismo.

Escenarios Alternativos

- *a. El jugador cancela el proceso de cambio de aceleración.

6.1.7. Caso de Uso: Activar o Desactivar Sonidos

Caso de Uso: Activar o Desactivar Sonidos

Precondiciones: El fichero de configuración de audio debe existir.

Postcondiciones: El usuario activará o desactivará el sistema de audio.

Descripción: Poner en funcionamiento o no el sistema de audio.

Resumen: El jugador quiere activar o desactivar el sistema de audio de *Laboon.*

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Opciones".
2. El sistema devuelve la pantalla con las distintas opciones disponibles.
3. El jugador selecciona el botón "Sonidos".
4. El sistema muestra las opciones del sistema de audio.

5. El jugador elige si activa o desactiva el sistema de audio.

6. El sistema guarda los cambios y actualiza el estado del sistema de audio.

Escenarios Alternativos

- *a. El jugador cancela el proceso de cambio del sistema de audio.

6.1.8. Caso de Uso: Activar o Desactivar Subtítulos

Caso de Uso: Activar o Desactivar Subtítulos

Precondiciones: El archivo de configuración de vídeo debe existir.

Postcondiciones: El usuario activará o desactivará el sistema de audio.

Descripción: Poner en funcionamiento o no el sistema de subtítulos.

Resumen: El jugador quiere activar o desactivar el sistema de subtítulos.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Opciones".

2. El sistema devuelve la pantalla con las distintas opciones disponibles.

3. El jugador selecciona el botón "Subtítulos".

4. El sistema muestra las opciones del sistema de subtítulos.

5. El jugador elige si activa o desactiva el sistema de subtítulos.

6. El sistema guarda los cambios y actualiza el estado del sistema de subtítulos.

Escenarios Alternativos

- *a. El jugador cancela el proceso del sistema de subtítulos.

Dentro de *Gestión Videojuego*, poseemos más casos de usos pero esta vez son todos dispares. Hay muchas acciones que son bastante similares por lo tanto vamos a mostrar todas las funcionalidades distintas que se pueden realizar durante la una partida en *Laboon*.

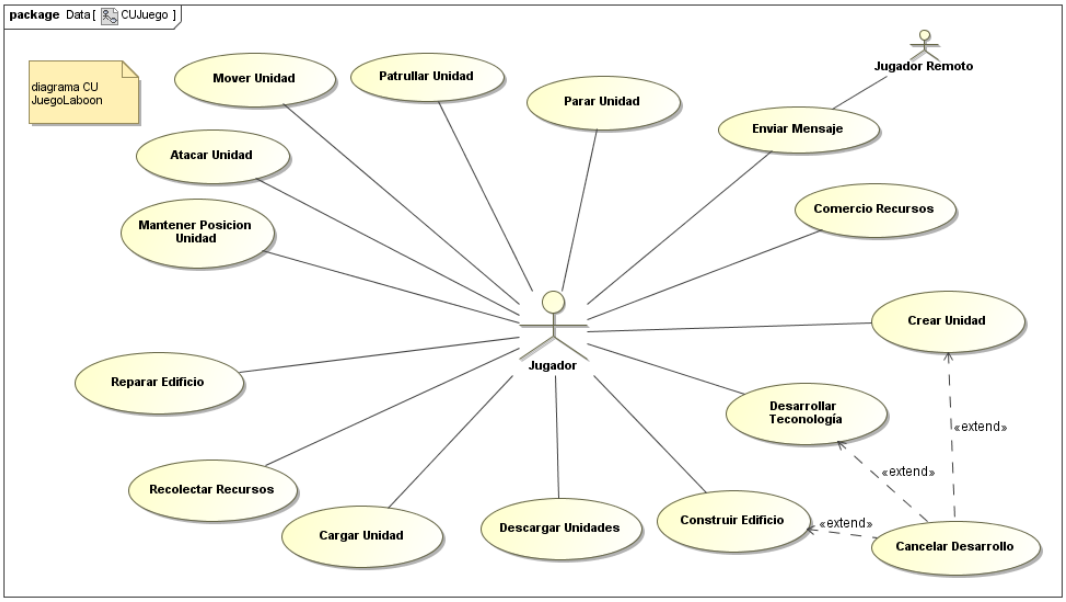

Figura 6.3: Diagrama de caso de uso del juego de *Laboon*.

6.1.9. Caso de Uso: Mover Unidad

Caso de Uso: Mover Unidad

Precondiciones: El jugador debe tener uno o varios vehículos propios seleccionados.

Postcondiciones: Las unidades comenzarán a moverse hasta el punto objetivo.

Descripción: El jugador mueve una o varias unidades.

Resumen: Un jugador quiere mover una o varias unidades desde una zona del mapa a un punto concreto.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón “Mover” o pulsa V.
2. El sistema cambia el puntero por una mira dando a entender que se ha aceptado la acción de mover.
3. El jugador selecciona el punto objetivo donde mover.
4. El sistema da las órdenes de mover a todos los vehículos seleccionados.
5. El sistema calcula los caminos de todos los vehículos.
6. El sistema empieza a mover los vehículos según el camino calculado.

Escenarios Alternativos

- *a. El jugador cancela el proceso de mover unidades.

6.1.10. Caso de Uso: Parar Unidad

Caso de Uso: Parar Unidad

Precondiciones: El jugador debe tener uno o varios vehículos propios seleccionados.

Postcondiciones: Las unidades dejarán de hacer lo que estén haciendo y pasarán a estado "Ocioso".

Descripción: El jugador para una o varias unidades.

Resumen: Un usuario quiere parar o hacer que dejen de actuar uno o varios vehículos.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Parar" o pulsa P.
2. El sistema da las órdenes de parar a todos los vehículos seleccionados.
3. El sistema hace que todos los vehículos dejen de moverse y/o actuar pasando todos a estado "Ocioso".

6.1.11. Caso de Uso: Atacar Unidad

Caso de Uso: Atacar Unidad

Precondiciones: El jugador debe tener uno o varios vehículos propios seleccionados.

Postcondiciones: Las unidades comenzarán a atacar a la unidad objetivo del jugador contrario.

Descripción: El jugador manda a atacar a una unidad enemiga.

Resumen: Un jugador quiere atacar con una o varias unidades desde una zona del mapa a una unidad concreto.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Atacar" o pulsa A.
2. El sistema cambia el puntero por una mira dando a entender que se ha aceptado la acción de atacar.
3. El jugador selecciona el punto objetivo donde atacar.

4. El sistema da las órdenes de atacar a todos los vehículos seleccionados.
5. Si una unidad está suficientemente cerca el sistema le mandará atacar, en caso contrario el sistema le mandará mover hasta llegar a rango de disparo.
6. El sistema calcula los caminos de todos los vehículos si es necesario.

Escenarios Alternativos

- *a. El jugador cancela el proceso de atacar unidades.

6.1.12. Caso de Uso: Mantener Posición Unidad

Caso de Uso: Mantener Posición Unidad

Precondiciones: El jugador debe tener uno o varios vehículos propios seleccionados.

Postcondiciones: Las unidades se quedarán acechando desde su posición actual.

Descripción: El jugador manda que sus vehículos mantengan su posición pase lo que pase.

Resumen: Un jugador quiere que sus unidades no se muevan bajo ningún concepto y que si alguna unidad entra en su rango de disparo ataquen.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Mantener Posición" o pulsa H.
2. El sistema da las órdenes de mantener posición a todos los vehículos seleccionados.
3. El sistema hace que todos los vehículos se paren comprobando periódicamente si hay amenazas cerca para atacar.

6.1.13. Caso de Uso: Patrullar Unidad

Caso de Uso: Patrullar Unidad

Precondiciones: El jugador debe tener un vehículo de ataque propio seleccionado.

Postcondiciones: La unidad seleccionada patrullará entre 2 puntos del mapa buscando enemigos.

Descripción: El jugador manda a patrullar a una unidad.

Resumen: Un jugador quiere que un vehículo de ataque patrulle entre 2 puntos en busca de amenazas cercanas.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón “Patrullar”.
2. El sistema cambia el puntero por una mira dando a entender que se ha aceptado la acción de patrullar.
3. El jugador selecciona el punto objetivo donde patrullar.
4. El sistema da la orden de patrullar al vehículo seleccionado.
5. El sistema calcula el camino del vehículo seleccionado.
6. Si una unidad está suficientemente cerca de una amenaza el sistema le mandará atacar, en caso contrario el vehículo seguirá patrullando indefinidamente.

Escenarios Alternativos

- *a. El jugador cancela el proceso de patrullar.

6.1.14. Caso de Uso: Construir Edificio

Caso de Uso: Construir Edificio

Precondiciones: El jugador debe tener un vehículo propio seleccionado de tipo “PeonConstructor”.

Postcondiciones: La unidad seleccionada iniciará el proceso de construcción del edificio que el jugador elija.

Descripción: El jugador manda a construir un edificio.

Resumen: Un jugador quiere construir un nuevo edificio en el tablero.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón “Construir” o pulsa C.
2. El sistema cambia el cuadro de mandos para mostrar los edificios que existen para construir.
3. El jugador selecciona el edificio que quiere construir.
4. El sistema comprueba que el jugador posee suficientes recursos para construir dicho edificio.
5. El sistema activa una sombra que simboliza el tamaño que ocupa el edificio y cambia el puntero a una mira dando a entender que ha entendido la petición.
6. El usuario elige donde quiere construir el edificio.
7. El sistema comprueba que la zona está limpia.
8. El sistema calcula la el camino hasta el punto donde se construirá el edificio.

9. El "PeonConstructor" se mueve hasta un punto cercano del sitio donde se construirá el edificio.
10. El sistema vuelve a comprobar que el usuario posee recursos suficientes para construir el edificio y que la zona está libre.
11. El sistema crea un nuevo edificio en el tablero, reduciendo los recursos del jugador que lo ha solicitado.

Escenarios Alternativos

- *a. El jugador cancela el proceso de reparar un edificio.
- 4b. El jugador no posee suficientes recursos para construir el edificio devolviendo un mensaje de error.
- 7b. La zona de construcción está ocupada y no se puede construir en ella, el sistema devuelve un mensaje de error.
- 10b. El jugador no posee suficientes recursos para construir el edificio devolviendo un mensaje de error.
- 10c. La zona de construcción está ocupada y no se puede construir en ella, el sistema devuelve un mensaje de error.

6.1.15. Caso de Uso: Reparar Edificio

Caso de Uso: Reparar Edificio

Precondiciones: El jugador debe tener un vehículo propio seleccionado de tipo "PeonConstructor".

Postcondiciones: La unidad seleccionada iniciará el proceso de reparación del edificio que el jugador elija.

Descripción: El jugador manda a reparar un edificio.

Resumen: Un jugador quiere reparar un edificio propio del tablero.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Reparar" o pulsa R.
2. El sistema cambia puntero por una mira dando a entender que acepta la petición de reparar.
3. El jugador selecciona el edificio que quiere reparar.
4. El sistema comprueba que el edificio es del propio jugador.
5. El sistema calcula la el camino hasta el punto donde se encuentra el edificio.
6. El "PeonConstructor" se mueve hasta un punto cercano del sitio donde reparará el edificio.

7. El sistema da la orden al "PeonConstructor" de reparar el edificio.
8. El "PeonConstructor" repara el edificio objetivo.

Escenarios Alternativos

- *a. El jugador cancela el proceso de reparar un edificio.
- 4b. El edificio no es propio y el peón atacará al edificio enemigo.
- 8b. El jugador no posee suficientes recursos para seguir reparando el edificio y se queda en estado ocioso.

6.1.16. Caso de Uso: Recolectar Recursos

Caso de Uso: Recolectar Recursos

Precondiciones: El jugador debe tener un vehículo propio seleccionado de tipo "PeonLeñador", "PeonMinero" o "BarcoPeon".

Postcondiciones: La unidad seleccionada iniciará el proceso de recolección de recursos en un punto establecido.

Descripción: El jugador manda a una unidad a recolectar recursos.

Resumen: Un jugador quiere recolectar recursos del tablero.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Recolectar" o pulsa G.
2. El sistema cambia puntero por una mira dando a entender que acepta la petición de recolectar.
3. El jugador selecciona punto del mapa donde recolectar.
4. El sistema comprueba que la zona es una zona con recursos y dichos recursos son los que recolecta la unidad seleccionada.
5. El sistema calcula la el camino hasta el punto donde se recolectarán los recursos.
6. El vehículo seleccionado se mueve hasta un punto que se le ha marcado.
7. El sistema manda al vehículo a recolectar el recurso sobre el que se encuentra.

Escenarios Alternativos

- *a. El jugador cancela el proceso de recolectar recursos.
- 4b. El recurso asignado no se corresponde con el vehículo seleccionado.
- 4c. La zona seleccionada no posee recursos.

6.1.17. Caso de Uso: Cargar Unidad

Caso de Uso: Cargar Unidad

Precondiciones: El jugador debe tener un vehículo propio seleccionado de tipo "TanqueTransporte".

Postcondiciones: La unidad seleccionada cargará a la unidad propia objetivo.

Descripción: El jugador manda a la unidad seleccionada cargar a la unidad objetivo.

Resumen: Un jugador quiere introducir una unidad en el transporte.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Cargar Unidad" o pulsa L.
2. El sistema cambia puntero por una mira dando a entender que acepta la petición de carga.
3. El jugador selecciona una unidad a la que cargar.
4. El sistema comprueba que la unidad es propia y que tiene el tamaño adecuado.
5. El sistema calcula la el camino hasta el punto donde se encuentra la unidad.
6. El vehículo seleccionado se mueve hasta un punto que se le ha marcado.
7. El sistema manda al vehículo a cargar la unidad objetivo.
8. La unidad objetivo desaparece del tablero y queda dentro del transporte.

Escenarios Alternativos

- *a. El jugador cancela el proceso de carga de unidades.
- 4b. La unidad seleccionada no es propia y el vehículo seleccionado ataca a la unidad objetivo.
- 4c. La unidad seleccionada es más grande de lo permitido y no se introduce en el transporte devolviendo un mensaje de error.

6.1.18. Caso de Uso: Descargar Unidades

Caso de Uso: Descargar Unidades

Precondiciones: El jugador debe tener un vehículo propio seleccionado de tipo "TanqueTransporte".

Postcondiciones: La unidad seleccionada descargará todas las unidades que haya en su interior.

Descripción: El jugador manda a la unidad seleccionada descargar las unidades que están dentro.

Resumen: Un jugador quiere extraer todas las unidades del transporte.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Descargar Unidades" o pulsa U.
2. El sistema para el vehículo seleccionado.
3. El sistema comprueba que hay espacio para extraer las unidades.
4. El sistema manda al transporte a descargar sus unidades.
5. Las unidades que están dentro del transporte aparecen en el tablero.

Escenarios Alternativos

- *a. El jugador cancela el proceso de descarga de unidades.
- 3b. No hay espacio para extraer más unidades, el vehículo seleccionado se queda ocioso y el sistema devuelve un mensaje de error.

6.1.19. Caso de Uso: Desarrollar Tecnología

Caso de Uso: Desarrollar Tecnología

Precondiciones: El jugador debe tener un edificio propio seleccionado de tipo "LaboratorioInvestigacion".

Postcondiciones: La unidad seleccionada comenzará a desarrollar la tecnología especificada.

Descripción: El jugador manda a desarrollar una nueva tecnología.

Resumen: Un jugador quiere evolucionar sus unidades a través del desarrollo de una nueva tecnología.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona la tecnología que quiera desarrollar.
2. El sistema comprueba que el jugador posea recursos suficientes.

3. El sistema avisa al edificio para que comience a desarrollar la tecnología.

Escenarios Alternativos

- *a. El jugador cancela el proceso de desarrollo de tecnología
- 2b. No hay recursos suficientes, el sistema devuelve mensaje de error.

6.1.20. Caso de Uso: Crear Unidad

Caso de Uso: Crear Unidad

Precondiciones: El jugador debe tener seleccionado un edificio propio que cree unidades.

Postcondiciones: La unidad seleccionada comenzará a desarrollar la unidad especificada.

Descripción: El jugador manda a construir una nueva unidad.

Resumen: Un jugador quiere incrementar la cantidad de unidades en el tablero.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona la unidad a desarrollar.
2. El sistema comprueba que el jugador posea recursos suficientes.
3. El sistema comprueba que el jugador tenga población libre suficiente para crear dicha unidad.
4. El sistema avisa al edificio para que comience a desarrollar la unidad.
5. El edificio construye la unidad objetivo.
6. El sistema comprueba que hay espacio en el tablero para añadir la unidad.
7. El sistema añade una unidad en el tablero.

Escenarios Alternativos

- *a. El jugador cancela el proceso de desarrollo de unidad
- 2b. No hay recursos suficientes, el sistema devuelve mensaje de error.
- 3b. No población suficiente, el sistema devuelve mensaje de error.
- 5b. No sitio para desplegar, el sistema devuelve mensaje de error y devuelve los recursos al jugador.

6.1.21. Caso de Uso: Cancelar Desarrollo

Caso de Uso: Cancelar Desarrollo

Precondiciones: El jugador debe tener seleccionado un edificio propio que cree unidades y/o tecnologías y dicho edificio debe estar en mitad de un desarrollo.

Postcondiciones: El edificio seleccionado dejará de desarrollar la unidad/tecnología actual.

Descripción: El jugador cancela un desarrollo.

Resumen: Un jugador quiere dejar de desarrollar una unidad o una tecnología.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador selecciona el botón "Cancelar Desarrollo".
2. El sistema cancela el desarrollo y devuelve una parte de los recursos invertidos al jugador.

6.1.22. Caso de Uso: Comercio Recursos

Caso de Uso: Comercio Recursos

Precondiciones: El jugador debe tener seleccionado un edificio de tipo "Mercado".

Postcondiciones: Se hará un trueque de recursos.

Descripción: El jugador cambia metal o madera por el recurso contrario.

Resumen: Un jugador quiere conseguir rápidamente recursos de un tipo a cambio de entregar recursos del tipo contrario.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador pulsa el botón del recurso que necesita.
2. El sistema comprueba que el jugador tiene recursos suficientes del tipo contrario.
3. El sistema extrae los recursos del tipo entregado del jugador y le entrega 100 unidades del recurso solicitado.
4. El sistema actualiza los índices de intercambio

Escenarios Alternativos

- 2a. El jugador no tiene suficientes recursos del tipo solicitado para hacer el intercambio.

6.1.23. Caso de Uso: Enviar Mensaje

Caso de Uso: Enviar Mensaje

Precondiciones: El jugador debe estar en una partida tipo **Multijugador**.

Postcondiciones: Se enviará un mensaje al jugador contrario.

Descripción: El jugador manda un mensaje de texto al jugador enemigo a través del chat.

Resumen: Un jugador quiere decirle algo al jugador enemigo.

Actores: Jugador (Principal)

Escenario Principal

1. El jugador pulsa la teclar INTRO.
2. El sistema abre la ventana de chat.
3. El jugador introduce su mensaje y vuelve a pulsar INTRO.
4. El sistema empaqueta el mensaje y lo envía al jugador remoto.
5. El sistema del jugador remoto desempaqueta el mensaje y se lo muestra por pantalla.

Escenarios Alternativos

- *a. El jugador cancela el proceso de envío de mensaje por chat.

6.2. Modelo Conceptual

El modelado conceptual de los datos nos permite modelar los requisitos descritos usando estructuras de dates y relaciones entre ellos.

La técnica que utilizaremos para realizar y presentar el modelado es un diagrama de clases.

Este modelado incluye:

- Clases.
- Asociaciones.
- Atributos.
- Restricciones de seguridad.

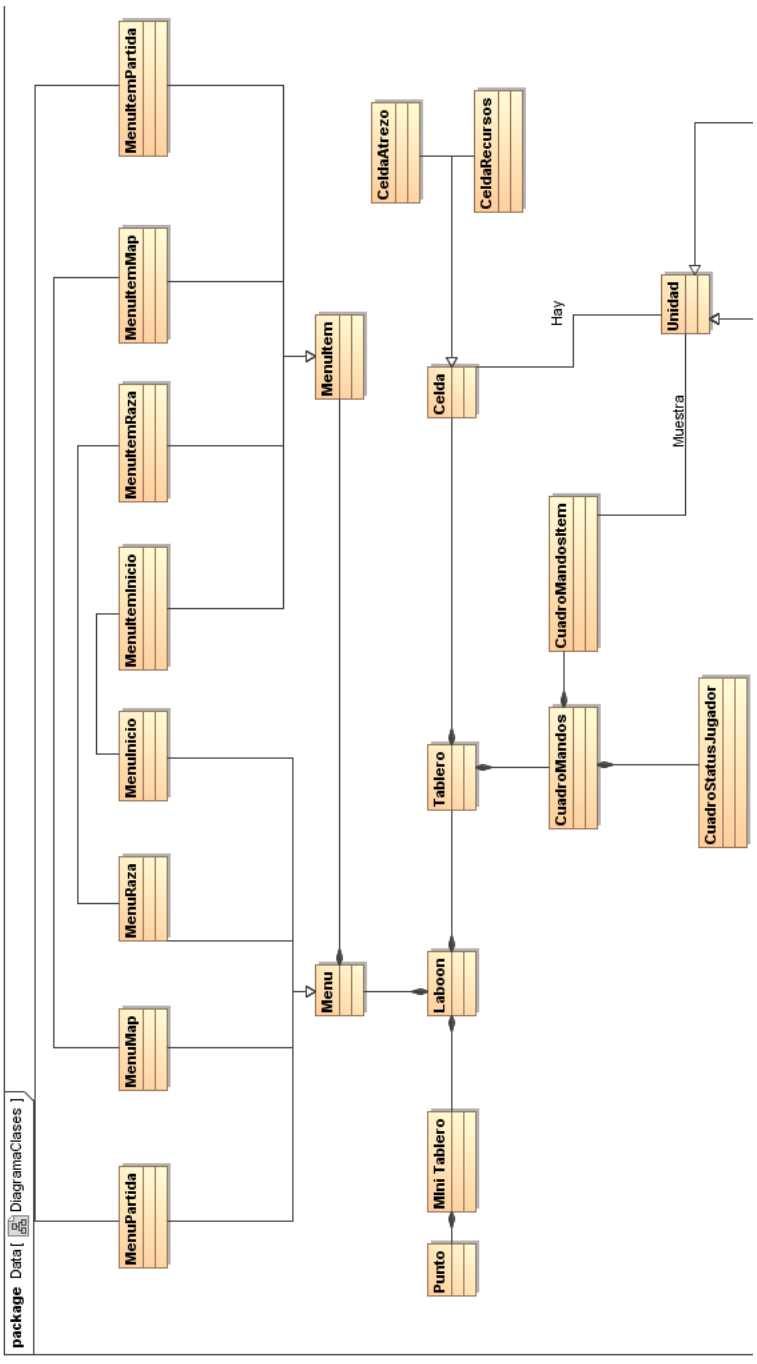

Figura 6.4: Diagrama conceptual de clases (Parte1).

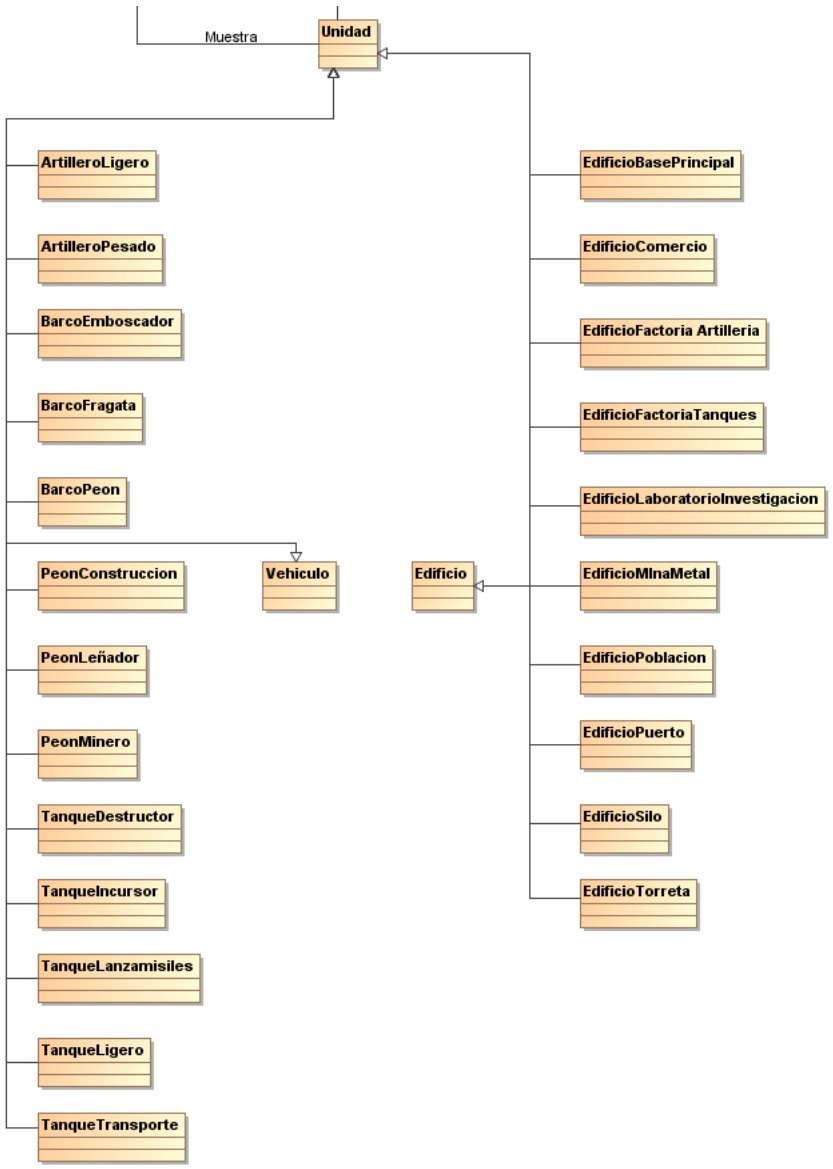

Figura 6.5: Diagrama conceptual de clases (Parte2).

6.3. Modelo de Comportamiento del sistema

El modelado del comportamiento del sistema nos permite obtener la especificación del comportamiento del mismo. Las técnicas que utilizaremos para el modelado serán los diagramas de secuencia y los contratos de las operaciones.

Los diagramas de comportamiento muestran el comportamiento de eventos que producen los actores y el sistema. Nos ayudan a identificar las operaciones

del sistema.

Los contratos nos permiten describir el efecto de las operaciones del sistema.

6.3.1. Modelo de Comportamiento: Un Jugador

Diagrama de comportamiento:

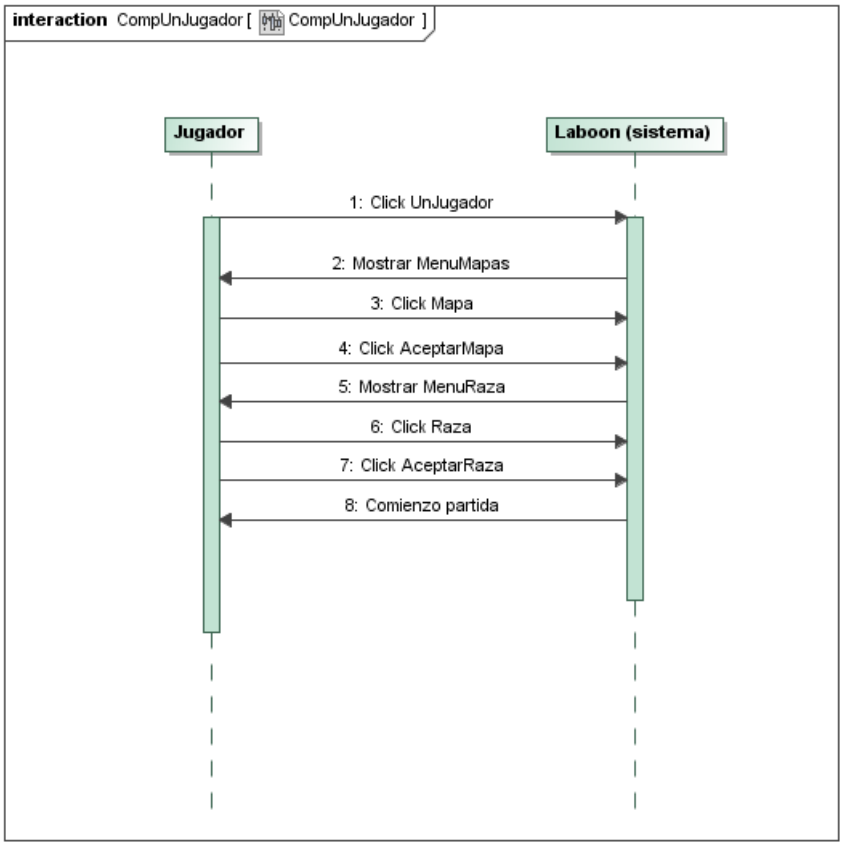

Figura 6.6: Diagrama Comportamiento: Un Jugador.

Contratos Operaciones

Operación: Click UnJugador

Responsabilidad: Comenzar el proceso de una partida contra el sistema.

Precondiciones: Ninguna

Postcondiciones:

- El sistema crea al jugador local y prepara el menú de mapas.

Operación: Mostrar MenuMapas

Responsabilidad: Mostrar los mapas disponibles en el sistema.

Precondiciones: Debe existir al menos un mapa válido en el directorio correspondiente.

Postcondiciones:

- El jugador contempla una lista de mapas válidos donde jugar.

Operación: Click Mapa

Responsabilidad: Seleccionar un mapa dentro de la lista de mapas disponibles.

Precondiciones: Debe existir al menos un mapa disponible.

Postcondiciones:

- El mapa queda seleccionado.

Operación: Click AceptarMapa

Responsabilidad: Crear las estructuras necesarias para poder representar el mapa seleccionado.

Precondiciones: Debe haber un mapa seleccionado previamente de la lista.

Postcondiciones:

- Se almacena el mapa seleccionado y se abren los ficheros correspondientes a dicho mapa para obtener la información correspondiente.

Operación: Mostrar MenuRaza

Responsabilidad: Mostrar los bandos disponibles con las que el jugador puede jugar.

Precondciones: Ninguna

Postcondiciones:

- El jugador contempla el conjunto de bandos disponibles.

Operación: Click Raza

Responsabilidad: Seleccionar una bandos de las disponibles en el sistema.

Precondiciones: Debe existir al menos una bando disponible.

Postcondiciones:

- El bando queda seleccionada.

Operación: Click AceptarRaza

Responsabilidad: Crear las estructuras necesarias para poder representar el bando seleccionado.

Precondiciones: Debe existir una bando seleccionado.

Postcondiciones:

- Se almacena la bando seleccionado y se abren los ficheros correspondientes a dicho bando para obtener la información correspondiente.

Operación: Comienzo Partida

Responsabilidad: Iniciar la partida en modo "Un Jugador".

Precondiciones: Deben haberse cumplido correctamente todas las operaciones anteriores.

Postcondiciones:

- Se terminan de construir las estructuras necesarias para el uso del mapa seleccionado.
- Se terminan de construir las estructuras necesarias para el uso de la bando seleccionado.
- Se inicia una partida en modo "Un Jugador" del jugador contra el sistema.
- Al sistema se le asigna una bando al azar.
- Se abandona el menú del sistema.

6.3.2. Modelo de Comportamiento: Crear Partida Multijugador

Diagrama de comportamiento:

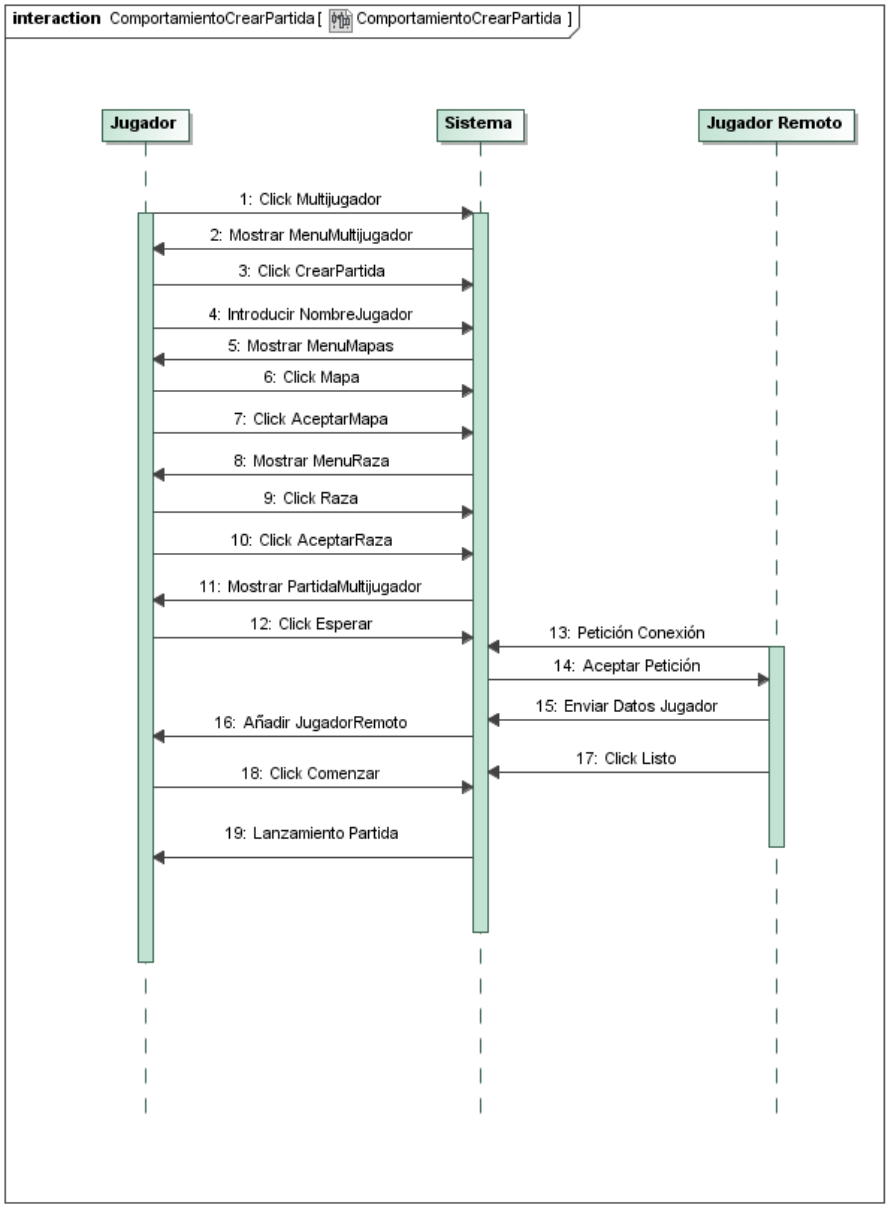

Figura 6.7: Diagrama Comportamiento: Crear Partida Multijugador.

Contratos Operaciones

Operación: Click Multijugador

Responsabilidad: Comenzar el proceso de crear una partida para jugar contra un jugador remoto.

Precondiciones: Ninguna

Postcondiciones:

- Se crean las estructuras necesarias y se inicia el proceso de creación de una partida multijugador.

Operación: Mostrar MenuMultijugador

Responsabilidad: Mostrar al jugador los distintos modos en los que se puede realizar una partida multijugador.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla las distintas formas de empezar una partida multijugador.

Operación: Click CrearPartida

Responsabilidad: Comenzar una partida en modo "Multijugador" siendo el sistema actual el *host* de la partida.

Precondiciones: Ninguna.

Postcondiciones:

- Se crean las estructuras para una partida "Multijugador" comprobando que el sistema posee conectividad hacia el exterior.

Operación: Introducir NombreJugador

Responsabilidad: Asignar al jugador local el nombre escrito por el jugador.

Precondiciones: Ninguna.

Postcondiciones:

- Se almacena el nombre introducido por el jugador y se le asigna al jugador local.

Operación: Mostrar MenuMapas

Responsabilidad: Mostrar los mapas disponibles en el sistema.

Precondiciones: Debe existir al menos un mapa válido en el directorio correspondiente.

Postcondiciones:

- El jugador contempla una lista de mapas válidos donde jugar.

Operación: Click Mapa

Responsabilidad: Seleccionar un mapa dentro de la lista de mapas disponibles.

Precondiciones: Debe existir al menos un mapa disponible.

Postcondiciones:

- El mapa queda seleccionado.

Operación: Click AceptarMapa

Responsabilidad: Crear las estructuras necesarias para poder representar el mapa seleccionado.

Precondiciones: Debe haber un mapa seleccionado previamente de la lista.

Postcondiciones:

- Se almacena el mapa seleccionado y se abren los ficheros correspondientes a dicho mapa para obtener la información correspondiente.

Operación: Mostrar MenuRaza

Responsabilidad: Mostrar los bandos disponibles con las que el jugador puede jugar.

Precondciones: Ninguna

Postcondiciones:

- El jugador contempla el conjunto de bandos disponibles.

Operación: Click Raza

Responsabilidad: Seleccionar un bando de los disponibles en el sistema.

Precondiciones: Debe existir al menos un bando disponible.

Postcondiciones:

- El bando queda seleccionado.

Operación: Click AceptarRaza

Responsabilidad: Crear las estructuras necesarias para poder representar el bando seleccionada.

Precondiciones: Debe existir un bando seleccionado.

Postcondiciones:

- Se almacena el bando seleccionado y se abren los ficheros correspondientes a dicho bando para obtener la información correspondiente.

Operación: Mostrar PartidaMultijugador

Respondabilidad: Ofrecer un menú donde el jugador pueda aceptar peticiones de conexión de jugadores remotos.

Precondiciones: Ninguna

Postcondiciones:

- El jugador contempla las distintas opciones del menú actual.

Operación: Click Esperar

Precondiciones: EL sistema debe poseer conectividad externa.

Postoncidicones:

- El sistema se queda a la escucha de peticiones externas durante un tiempo determinado por el mismo sistema.

Operación: Petición Conexión

Responsabilidad: El jugador remoto solicita a nuestro sistema acceder a la partida que se va a comenzar.

Precondiciones: El jugador remoto debe haber introducido correctamente la IP del sistema actual.

Postcondiciones:

- El sistema recibe los datos necesarios para evaluar la petición externa.

Operación: Aceptar Petición

Responsabilidad: Enviar un mensaje de aceptación de la solicitud de un jugador remoto.

Precondiciones: Un jugador remoto debe haber solicitado una petición de unirse a la partida actual mientras el sistema se encontraba en escucha.

Postcondiciones:

- El jugador remoto tiene permiso para unirse a la partida en espera del sistema.

Operación: Enviar Datos Jugador

Responsabilidad: El jugador remoto envía los datos de su jugador local.

Precondiciones: El jugador remoto debe enviar los datos esperados por el sistema.

Postcondiciones:

- Se reciben los datos del jugador remoto para añadirlo en el sistema.

Operación: Añadir JugadorRemoto

Responsabilidad: Se añade un jugador remoto en el sistema.

Precondiciones: No debe existir un jugador remoto en el sistema actualmente.

Postcondiciones:

- Se crea un jugador remoto en el sistema.
- Se actualiza la lista de jugadores en espera.

Operación: Click Listo

Responsabilidad: El jugador remoto envía una señal de que se encuentra listo para comenzar la partida.

Precondiciones: El jugador no se debe encontrar listo.

Postondicones:

- El jugador remoto pasa a estado "Listo".
- Se refleja el cambio del estado del jugador en la pantalla de espera.

Operación: Click Comenzar

Responsabilidad: El jugador da paso a comenzar la partida multijugador.

Precondiciones: El jugador remoto debe encontrarse en estado "Listo".

Postcondiciones:

- El jugador local avisa al jugador remoto que la partida va a comenzar.
- El jugador remoto carga las estructuras necesarias para comenzar la partida.

Operación: Lanzamiento Partida

Responsabilidad: Comenzar la partida en modo multijugador.

Precondiciones: Se deben haber cumplido todos los pasos previos de forma satisfactoria.

Postcondiciones:

- Se terminan de construir las estructuras necesarias para el uso del mapa seleccionado.
- Se terminan de construir las estructuras necesarias para el uso de la bando seleccioando.
- Se inicia una partida en modo "Multijugador" del jugador contra el jugador remoto.
- Se abandona el menú del sistema.

6.3.3. Modelo de Comportamiento: Unirse a Partida Multijugador

Diagrama de comportamiento:

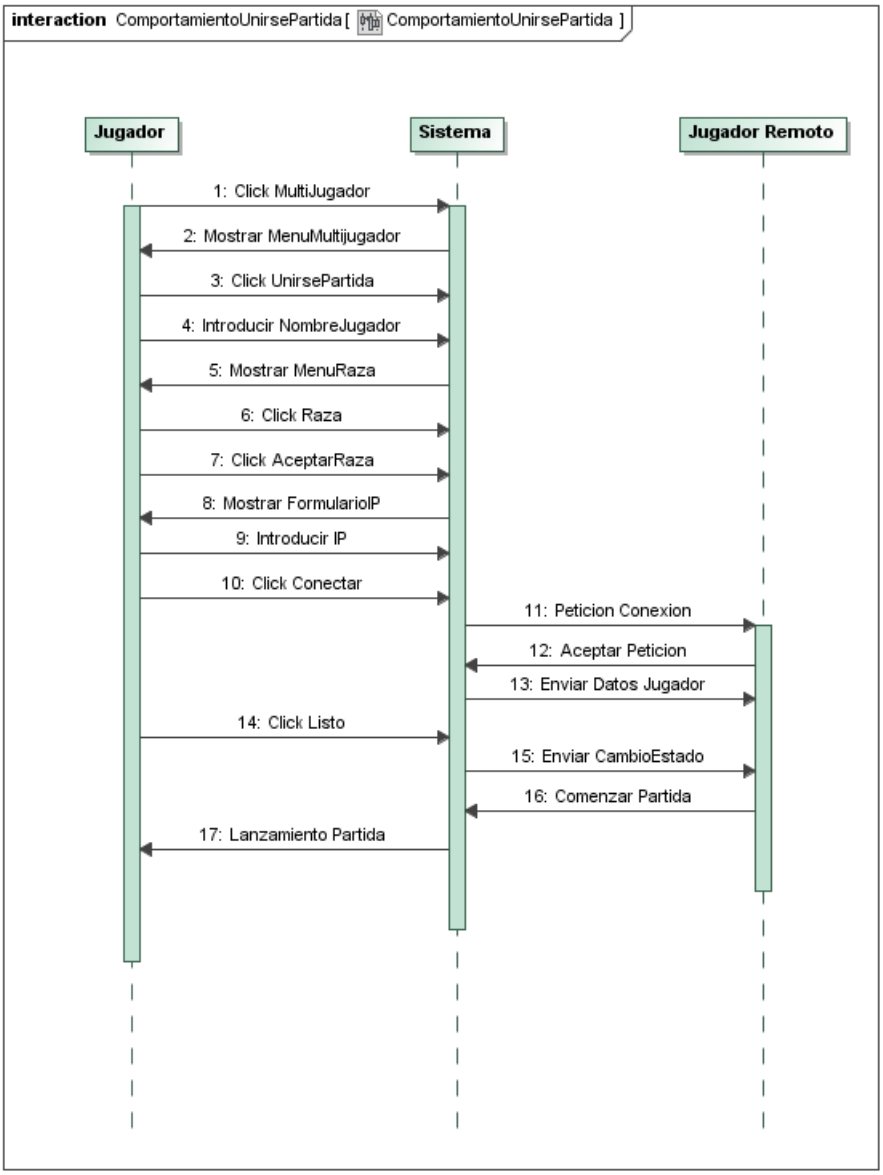

Figura 6.8: Diagrama Comportamiento: Unirse a Partida Multijugador.

Contratos Operaciones

Operación: Click Multijugador

Responsabilidad: Comenzar el proceso de crear una partida para jugar contra un jugador remoto.

Precondiciones: Ninguna

Postcondiciones:

- Se crean las estructuras necesarias y se inicia el proceso de creación de una partida multijugador.

Operación: Mostrar MenuMultijugador

Responsabilidad: Mostrar al jugador los distintos modos en los que se puede realizar una partida multijugador.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla las distintas formas de empezar una partida multijugador.

Operación: Click UnirsePartida

Responsabilidad: Comenzar una partida en modo "Multijugador" siendo el sistema actual el *cliente* de la partida.

Precondiciones: Ninguna

Postcondiciones:

- Se crean las estructuras para una partida "Multijugador" comprobando que el sistema posee conectividad hacia el exterior.

Operación: Introducir NombreJugador

Responsabilidad: Asignar al jugador local el nombre escrito por el jugador.

Precondiciones: Ninguna.

Postcondiciones:

- Se almacena el nombre introducido por el jugador y se le asigna al jugador local.

Operación: Mostrar MenuRaza

Responsabilidad: Mostrar los bandos disponibles con los que el jugador puede jugar.

Precondciones: Ninguna

Postcondiciones:

- El jugador contempla el conjunto de bandos disponibles.

Operación: Click Raza

Responsabilidad: Seleccionar un bando de los disponibles en el sistema.

Precondiciones: Debe existir al menos un bando disponible.

Postcondiciones:

- El bando queda seleccionado.

Operación: Click AceptarRaza

Responsabilidad: Crear las estructuras necesarias para poder representar el bando seleccionado.

Precondiciones: Debe existir un bando seleccionado.

Postcondiciones:

- Se almacena el bando seleccionado y se abren los ficheros correspondientes a dicho bando para obtener la información correspondiente.

Operación: Mostrar FormularioIP

Responsabilidad: Mostrar al jugador donde debe introducir la IP del jugador remoto que ha creado la partida.

Precondiciones: Debe existir un jugador remoto con una IP asignada y conectividad directa o indirecta hacia nuestro sistema que haya creado una partida.

Postcondiciones:

- El jugador contempla el formulario para introducir la IP del jugador Remoto.

Operación: Click Conectar

Responsabilidad: Conectar al sistema de un jugador remoto para iniciar una partida multijugador.

Precondiciones: La IP introducida anteriormente debe coincidir con la del jugador remoto.

PostCondiciones:

- Se crean las estructuras de conexión.
- Se comprueba la conectividad de nuestro sistema con el jugador remoto.

Operación: Petición Conexion

Responsabilidad: El sistema solicita al jugador remoto poder entrar en la partida multijugador creada por él.

Precondiciones: El jugador remoto debe haber creado una partida en modo "Multijugador" y encontrarse en espera.

Postcondiciones:

- El jugador remoto tiene constancia de que hay una petición para conectarse y se dispondrá a evaluarla.

Operación: Aceptar Petición

Responsabilidad: El jugador remoto envía un mensaje de aceptación de la solicitud hecha por nuestro sistema.

Precondiciones: La petición enviada ha sido aceptada.

Postcondiciones:

- El sistema obtiene confirmación del jugador remoto.
- El jugador entra en la partida del jugador remoto.
- El jugador contempla el menú PartidaMultijugador.

Operación: Enviar Datos Jugador

Responsabilidad: El jugador remoto envía los datos de su jugador local.

Precondiciones: El jugador remoto debe enviar los datos esperados por el sistema.

Postcondiciones:

- Se reciben los datos del jugador remoto para añadirlo en el sistema.

Operación: Click Listo

Responsabilidad: El jugador envía una señal de que se encuentra listo para comenzar la partida.

Precondiciones: El jugador no se debe encontrar listo.

Postondicones:

- El jugador pasa a estado "Listo".
- El sistema prepara un paquete de información para avisar al jugador remoto del cambio de estado producido.

Operación: Enviar CambioEstado

Responsabilidad: El sistema debe dar constancia al jugador remoto que ha creado la partida del cambio de estado producido.

Precondiciones: Se ha debido producir un cambio de estado en el jugador.

Postcondiciones:

- El jugador remoto ve reflejado el cambio de estado del jugador.

Operación: Comenzar Partida

Responsabilidad: El jugador remoto envía un paquete de información avisando de que la partida se va a lanzar.

Precondiciones: El jugador debe encontrarse en estado "Listo".

Postcondiciones:

- El sistema interpreta el mensaje enviado por el jugador remoto y comienza a crear las estructuras necesarias para lanzar la partida.

Operación: Lanzamiento Partida

Responsabilidad: Comenzar la partida en modo multijugador.

Precondiciones: Se deben haber cumplido todos los pasos previos de forma satisfactoria.

Postcondiciones:

- Se terminan de construir las estructuras necesarias para el uso del mapa seleccionado.
- Se terminan de construir las estructuras necesarias para el uso del bando seleccionado.
- Se inicia una partida en modo "Multijugador" del jugador contra el jugador remoto.
- Se abandona el menú del sistema.

6.3.4. Modelo de Comportamiento: Cambiar Resolución

Diagrama de comportamiento:

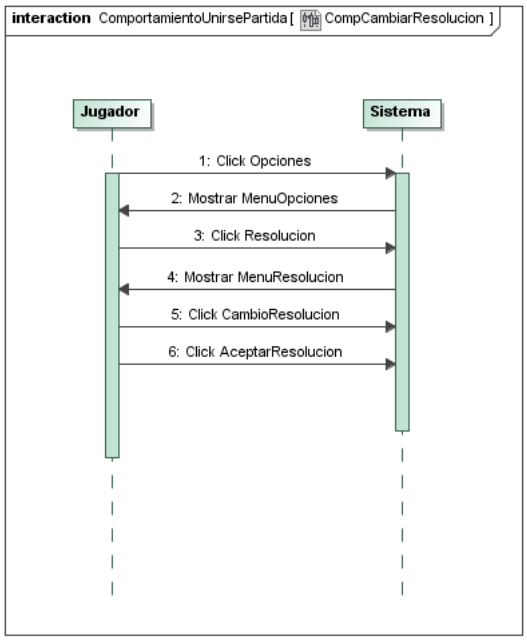

Figura 6.9: Diagrama Comportamiento: Cambiar Resolución del Sistema.

Contratos Operaciones

Operación: Click Opciones

Responsabilidad: Acceder al modo configuración del sistema.

Precondiciones: Ninguna.

Postoncidiones:

- El sistema se pone en modo configuración para que se puedan modificar sus parámetros.

Operación Mostrar MenuOpciones

Responsabilidad: Mostrar al usuario las distintas opciones configurables del sistema.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla todas las opciones disponibles a modificar.

Operación: Click Resolucion

Responsabilidad: Cargar las estructuras para modificar la resolución del sistema.

Precondiciones: Debe existir el fichero de configuración de video.

Postcondiciones:

- El sistema abre el fichero de configuración de vídeo para su posterior modificación.

Operación: Mostrar MenuResolucion

Responsabilidad: Mostrar al usuario las distintas resoluciones disponibles en el juego.

Precondiciones: Debe existir el fichero de configuración de vídeo.

Postcondiciones:

- El usuario contempla las distintas resoluciones que proporciona el sistema.

Operación: Click CambioResolucion

Responsabilidad: Elegir la resolución que el jugador desea.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El jugador elige la resolución que desea.

Operación Click AceptarResolucion

Responsabilidad: Guardar el cambio de resolución en el sistema.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El sistema modifica su resolución actual por la elegida por el jugador.
- El fichero de configuración de video se modifica.
- El sistema vuelve al menú.

6.3.5. Modelo de Comportamiento: Activar o Desactivar Aceleración Hardware

Diagrama de comportamiento:

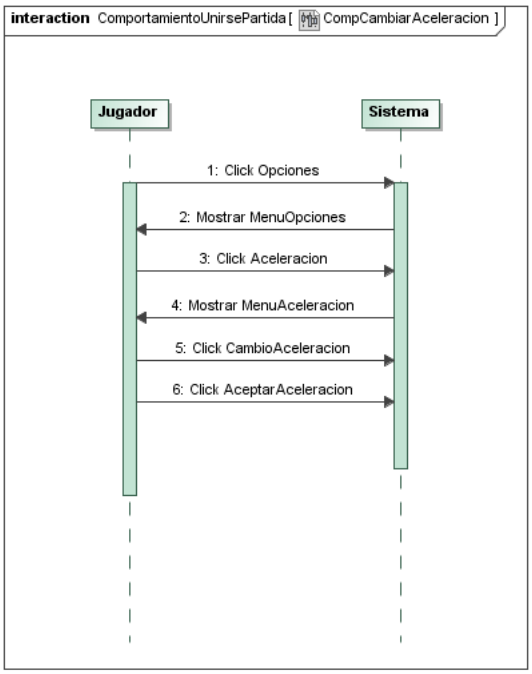

Figura 6.10: Diagrama Comportamiento: Cambiar Aceleración del Sistema.

Contratos Operaciones

Operación: Click Opciones

Responsabilidad: Acceder al modo configuración del sistema.

Precondiciones: Ninguna.

Postoncidiones:

- El sistema se pone en modo configuración para que se puedan modificar sus parámetros.

Operación Mostrar MenuOpciones

Responsabilidad: Mostrar al usuario las distintas opciones configurables del sistema.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla todas las opciones disponibles a modificar.

Operación: Click Aceleracion

Responsabilidad: Cargar las estructuras para modificar la aceleración del sistema.

Precondiciones: Debe existir el fichero de configuración de video.

Postcondiciones:

- El sistema abre el fichero de configuración de vídeo para su posterior modificación.

Operación: Mostrar MenuAceleracion

Responsabilidad: Mostrar al usuario las distintas aceleraciones disponibles en el juego.

Precondiciones: Debe existir el fichero de configuración de vídeo.

Postcondiciones:

- El usuario contempla las distintas aceleraciones que proporciona el sistema.

Operación: Click CambioAceleracion

Responsabilidad: Elegir la aceleración que el jugador desea.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El jugador elige la aceleración que desea.

Operación Click AceptarAceleracion

Responsabilidad: Guardar el cambio de aceleración en el sistema.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El sistema modifica su aceleración actual por la elegida por el jugador.
- El fichero de configuración de video se modifica.
- El sistema vuelve al menú.

6.3.6. Modelo de Comportamiento: Activar o Desactivar Sonidos

Diagrama de comportamiento:

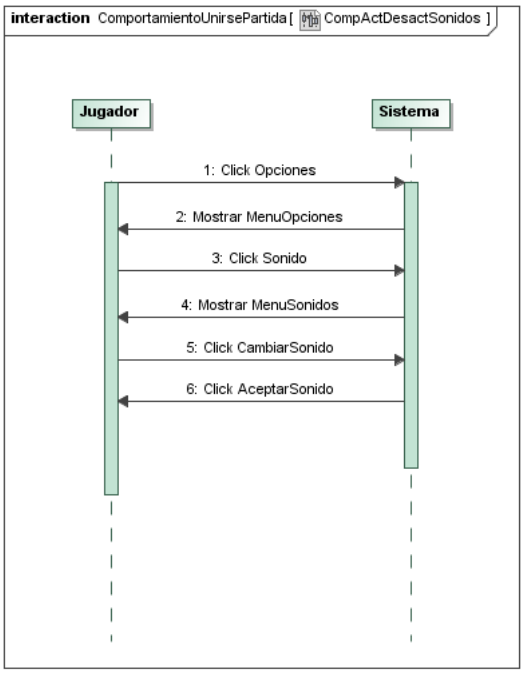

Figura 6.11: Diagrama Comportamiento: Activar o Desactivar Sistema de Audio.

Contratos Operaciones

Operación: Click Opciones

Responsabilidad: Acceder al modo configuración del sistema.

Precondiciones: Ninguna.

Postoncidiones:

- El sistema se pone en modo configuración para que se puedan modificar sus parámetros.

Operación Mostrar MenuOpciones

Responsabilidad: Mostrar al usuario las distintas opciones configurables del sistema.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla todas las opciones disponibles a modificar.

Operación: Click Sonidos

Responsabilidad: Cargar las estructuras para modificar el sistema de audio.

Precondiciones: Debe existir el fichero de configuración de audio.

Postcondiciones:

- El sistema abre el fichero de configuración de audio para su posterior modificación.

Operación: Mostrar MenuSonidos

Responsabilidad: Mostrar al usuario las opciones de activar o desactivar el sistema de audio.

Precondiciones: Debe existir el fichero de configuración de audio.

Postcondiciones:

- El usuario contempla las distintas opciones del sistema de audio.

Operación: Click CambiarSonido

Responsabilidad: Elegir si activar o desactivar el sistema de audio.

Precondiciones: El fichero de configuración de audio debe existir.

Postcondiciones:

- El sistema prepara las estructuras para modificar el sistema de audio.

Operación Click AceptarSonido

Responsabilidad: Guardar el cambio del sistema de audio.

Precondiciones: El fichero de configuración de audio debe existir.

Postcondiciones:

- El sistema modifica el sistema de audio.
- El fichero de configuración de audio se modifica.
- El sistema vuelve al menú.

6.3.7. Modelo de Comportamiento: Activar o Desactivar Subtítulos

Diagrama de comportamiento:

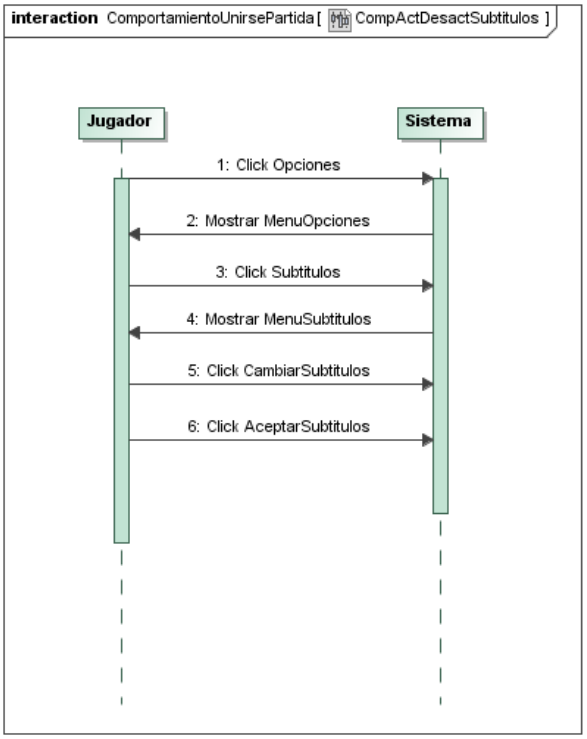

Figura 6.12: Diagrama Comportamiento: Activar o Desactivar Sistema de subtítulos.

Contratos Operaciones

Operación: Click Opciones

Responsabilidad: Acceder al modo configuración del sistema.

Precondiciones: Ninguna.

Postoncidiones:

- El sistema se pone en modo configuración para que se puedan modificar sus parámetros.

Operación Mostrar MenuOpciones

Responsabilidad: Mostrar al usuario las distintas opciones configurables del sistema.

Precondiciones: Ninguna.

Postcondiciones:

- El jugador contempla todas las opciones disponibles a modificar.

Operación: Click subtítulos

Responsabilidad: Cargar las estructuras para modificar el sistema de subtítulos.

Precondiciones: Debe existir el fichero de configuración de vídeo.

Postcondiciones:

- El sistema abre el fichero de configuración de video para su posterior modificación.

Operación: Mostrar MenuSubtitulos

Responsabilidad: Mostrar al usuario las opciones de activar o desactivar el sistema de subtítulos.

Precondiciones: Debe existir el fichero de configuración de vídeo.

Postcondiciones:

- El usuario contempla las distintas opciones del sistema de vídeo.

Operación: Click CambiarSubtitulos

Responsabilidad: Elegir si activar o desactivar el sistema de vídeo.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El sistema prepara las estructuras para modificar el sistema de subtítulos.

Operación Click AceptarSubtitulos

Responsabilidad: Guardar el cambio del sistema de subtítulos.

Precondiciones: El fichero de configuración de vídeo debe existir.

Postcondiciones:

- El sistema modifica el sistema de subtítulos.
- El fichero de configuración de vídeo se modifica.
- El sistema vuelve al menú.

6.3.8. Modelo de Comportamiento: Mover Unidad

Diagrama de comportamiento:

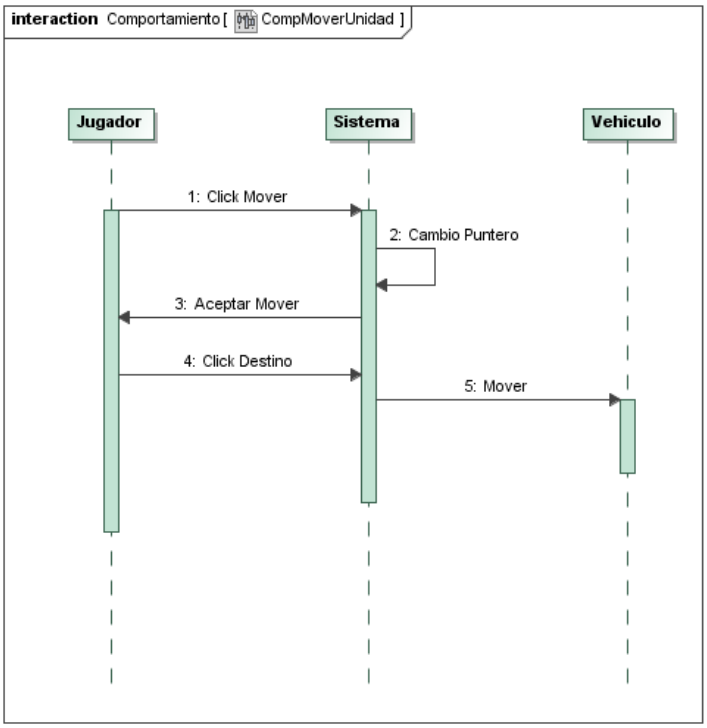

Figura 6.13: Diagrama Comportamiento: Mover Unidad.

Operación: Click Mover

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Mover".

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Mover

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el punto objetivo al cual deben moverse los vehículos.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema empieza a calcular el camino desde el punto origen al punto destino de los vehículos seleccionados.

Operación: Mover

Responsabilidad: El sistema indica a los vehículos el camino a recorrer.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- Los vehículos cambian su estado a "Mover" y los vehículos empiezan a moverse.

6.3.9. Modelo de Comportamiento: Parar Unidad

Diagrama de comportamiento:

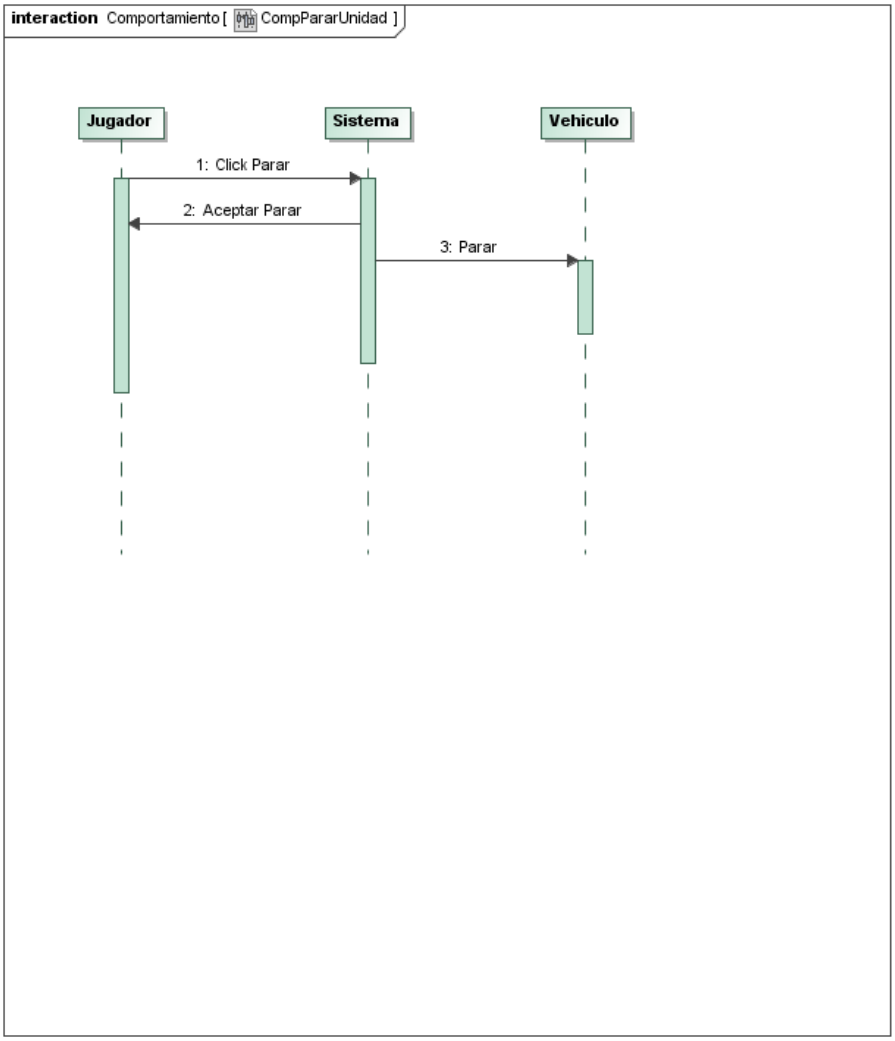

Figura 6.14: Diagrama Comportamiento: Parar Unidad.

Operación: Click Parar

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Parar".

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Aceptar Parar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Parar

Responsabilidad: El sistema indica a los vehículos que dejen de actuar.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- Los vehículos cambian su estado a "Ocioso" y se quedan quietos.

6.3.10. Modelo de Comportamiento: Mantener Posición Unidad

Diagrama de comportamiento:

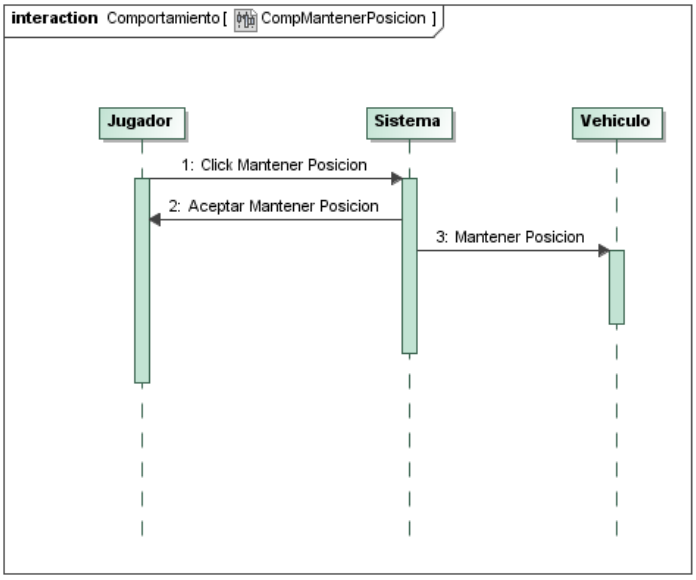

Figura 6.15: Diagrama Comportamiento: Mantener Posición.

Operación: Click Mantener Posición

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Mantener Posición".

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Aceptar Mantener Posición

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Mantener Posición

Responsabilidad: El sistema indica a los vehículos que mantengan su posición.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- Los vehículos cambian su estado a "Mantener Posición" y se quedan quietos.

6.3.11. Modelo de Comportamiento: Patrullar

Diagrama de comportamiento:

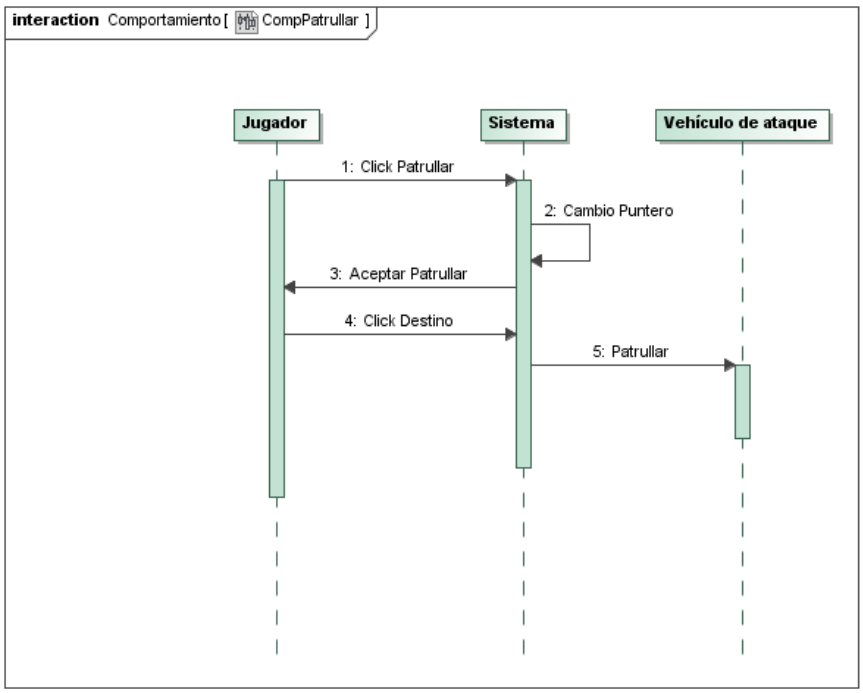

Figura 6.16: Diagrama Comportamiento: Patrullar.

Operación: Click Patrullar

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Patrullar".

Precondiciones: El jugador debe tener seleccionado un vehículo propio de tipo ataque.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado un vehículo propio de tipo ataque.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Patrullar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado un vehículo propio de tipo ataque.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el punto objetivo al cual deben patrullar los vehículos.

Precondiciones: El jugador debe tener seleccionado un vehículo propio de tipo ataque.

Postoncidiones:

- El sistema empieza a calcular el camino desde el punto origen al punto destino de los vehículos seleccionados.

Operación: Patrullar

Responsabilidad: El sistema indica a los vehículos el camino a recorrer.

Precondiciones: El jugador debe tener seleccionado un vehículo propio de tipo ataque.

Postoncidiones:

- El vehículo cambia su estado a "Patrullar" empieza a moverse.

6.3.12. Modelo de Comportamiento: Atacar Unidad

Diagrama de comportamiento:

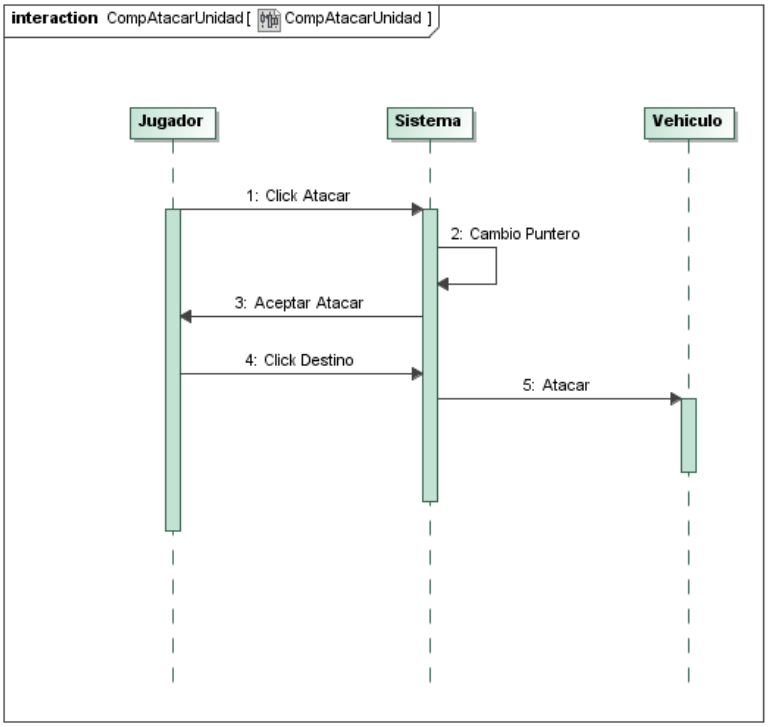

Figura 6.17: Diagrama Comportamiento: Atacar Unidad.

Operación: Click Atacar

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Atacar".

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Atacar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el punto objetivo al cual deben atacar los vehículos.

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- El sistema empieza a calcular las trayectorias de los disparos entre los vehículos origen y el objetivo, o si están muy lejos empezará a calcular los caminos a recorrer.

Operación: Atacar

Responsabilidad: El sistema indica a los vehículos a quién atacar

Precondiciones: El jugador debe tener seleccionado uno o varios vehículos propios.

Postoncidiones:

- Los vehículos cambian su estado a "Atacar" y los vehículos empiezan a atacar.

6.3.13. Modelo de Comportamiento: Reparar Edificio

Diagrama de comportamiento:

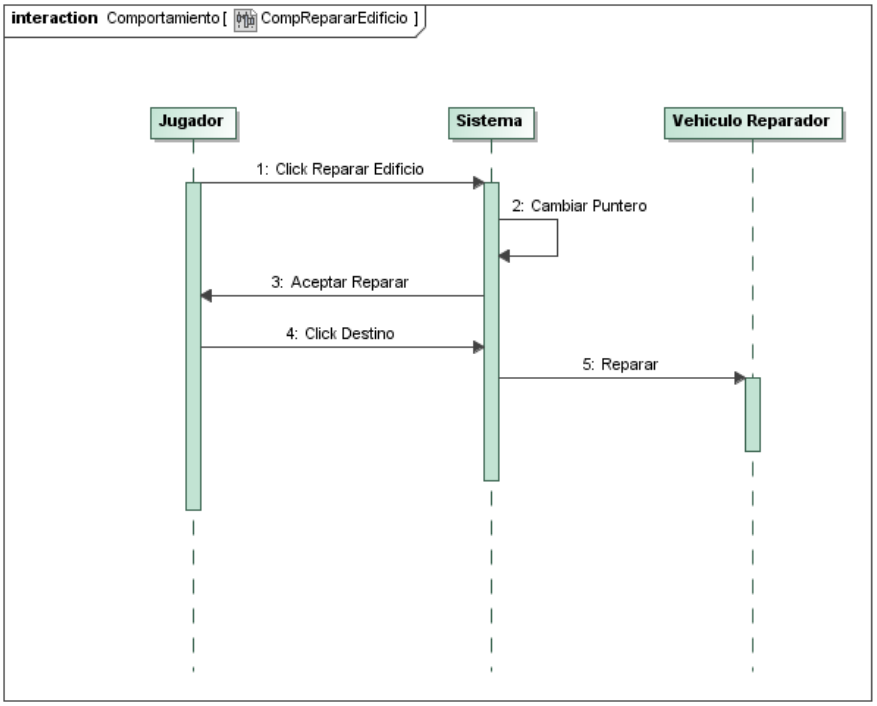

Figura 6.18: Diagrama Comportamiento: Reparar Edificio.

Operación: Click Reparar Edificio

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Reparar".

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo reparador.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo reparador.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Reparar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo reparador.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el punto objetivo que tiene que reparar el vehículo.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo reparador.

Postoncidiones:

- El sistema empieza a calcular el camino entre la unidad reparadora y el edificio.

Operación: Reparar

Responsabilidad: El sistema indica al vehículo qué debe reparar.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo reparador.

Postoncidiones:

- El vehículo cambia su estado a "Reparar" y el vehículo empieza a moverse.

6.3.14. Modelo de Comportamiento: Recolectar Recursos

Diagrama de comportamiento:

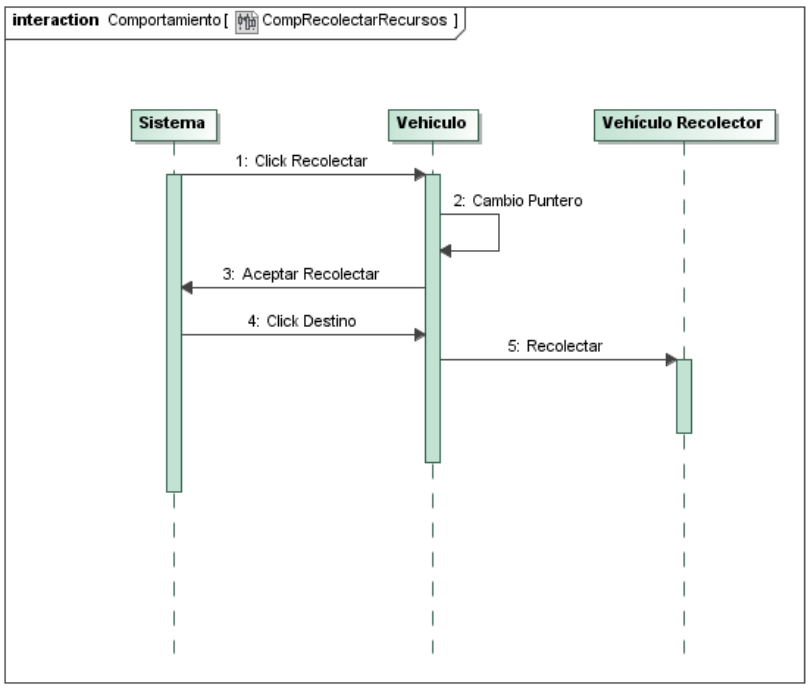

Figura 6.19: Diagrama Comportamiento: Recolectar Recursos.

Operación: Click Recolectar

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Recolectar".

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo recolector.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo recolector.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Recolectar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo recolector.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el punto objetivo donde tiene que recolectar el vehículo.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo recolector.

Postoncidiones:

- El sistema empieza a calcular el camino entre la unidad recolectora y la zona con recursos.

Operación: Recolectar

Responsabilidad: El sistema indica al vehículo qué debe recolectar.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo recolector.

Postoncidiones:

- El vehículo cambia su estado a "Recolectar" y el vehículo empieza a moverse.

6.3.15. Modelo de Comportamiento: Cargar Unidad

Diagrama de comportamiento:

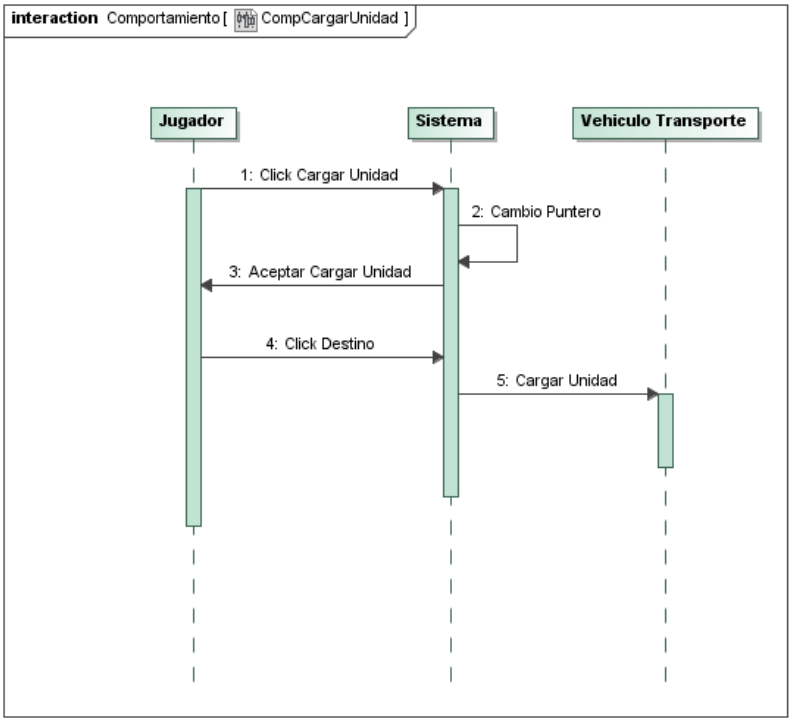

Figura 6.20: Diagrama Comportamiento: Cargar Unidad.

Operación: Click Cargar Unidad

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Cargar".

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El sistema cambia el puntero del ratón.

Operación: Aceptar Cargar Unidad

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica el objetivo que desea cargar.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El sistema comprueba que el vehículo es viable de ser cargado y comienza a calcular entre ambos.

Operación: Cargar Unidad

Responsabilidad: El sistema indica al vehículo qué vehículo debe cargar.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El vehículo cambia su estado a "Cargar" y el vehículo empieza a moverse.

6.3.16. Modelo de Comportamiento: Descargar Unidades

Diagrama de comportamiento:

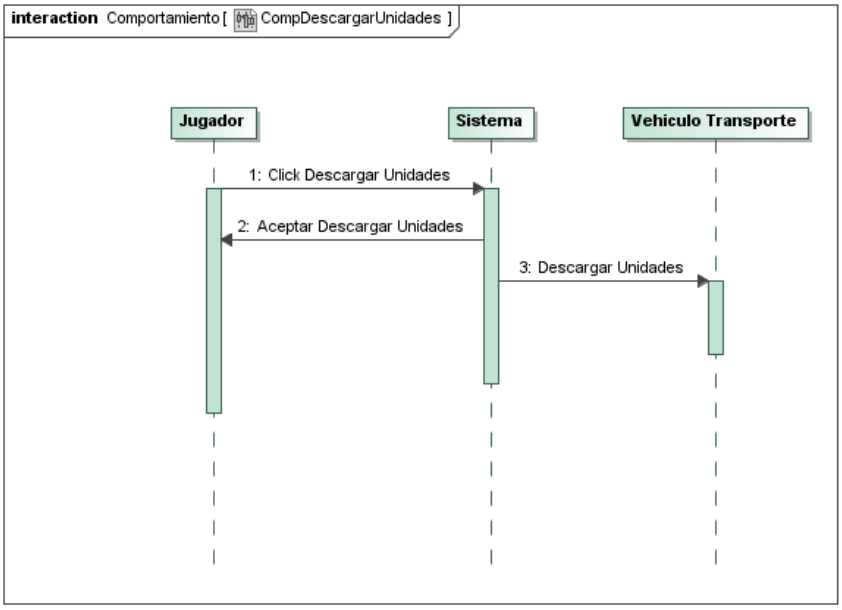

Figura 6.21: Diagrama Comportamiento: Descargar Unidades.

Operación: Click Descargar Unidades

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Descargar".

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Aceptar Descargar Unidades

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postoncidiones:

- El jugador vuelve a ganar el control.

Operación: Descargar Unidades

Responsabilidad: El sistema indica al vehículo que debe descargar sus unidades.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postconcidiones:

- El sistema comprueba que hay espacio para descargar las unidades que posee el transporte y le manda la orden de descargar.

6.3.17. Modelo de Comportamiento: Desarrollar Tecnología

Diagrama de comportamiento:

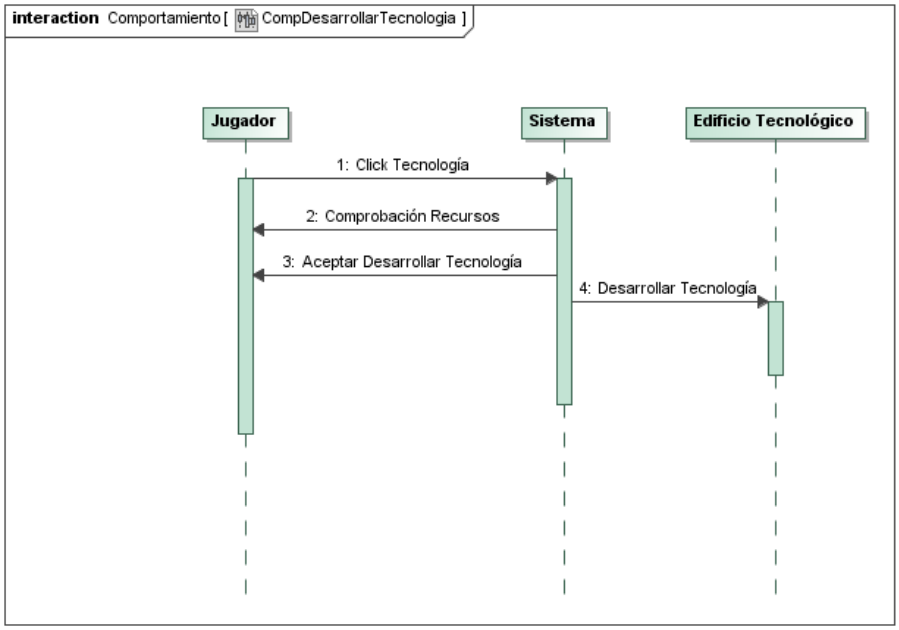

Figura 6.22: Diagrama Comportamiento: Desarrollar Tecnología.

Operación: Click Tecnología

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Desarrollar Tecnología".

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo investigador.

Postconcidiones:

- El sistema inicia el principio de cambio de estado del edificio

Operación: Comprobación Recursos

Responsabilidad: El sistema comprueba que el usuario posee suficientes recursos para desarrollar dicha tecnología.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo investigador.

Postconcidiones:

- El jugador es capaz de desarrollar la tecnología.

Operación: Aceptar Desarrollar Tecnología

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo investigador.

Postconcidiones:

- El jugador vuelve a ganar el control y el sistema decrementa los recursos del jugador.

Operación: Desarrollar Tecnología

Responsabilidad: El sistema avisa al edificio de tipo investigador sobre qué tecnología debe desarrollar.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo investigador.

Postconcidiones:

- El edificio comienza la investigación.

6.3.18. Modelo de Comportamiento: Crear Unidad

Diagrama de comportamiento:

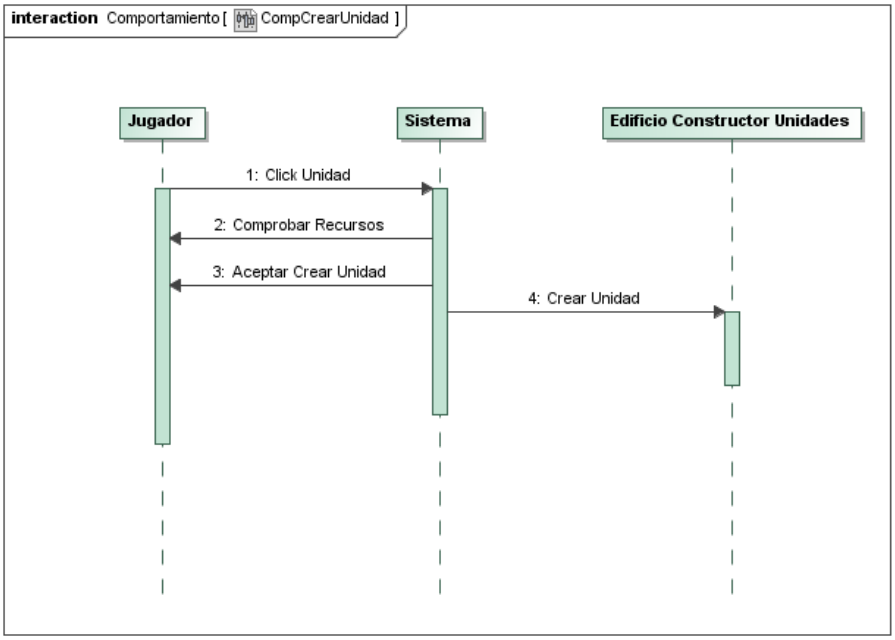

Figura 6.23: Diagrama Comportamiento: Crear Unidad.

Operación: Click Unidad

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Crear Unidad".

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades.

Postconcidiones:

- El sistema inicia el principio de cambio de estado del edificio.

Operación: Comprobación Recursos

Responsabilidad: El sistema comprueba que el usuario posee suficientes recursos para desarrollar dicha unidad.

Precondiciones: El jugador debe tener seleccionado un edificio propioque desarrolle unidades.

Postconcidiones:

- El jugador es capaz de desarrollar la unidad.

Operación: Aceptar Crear Unidad

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades.

Postconcidiones:

- El jugador vuelve a ganar el control y el sistema decrementa los recursos del jugador.

Operación: Crear Unidad

Responsabilidad: El sistema avisa al edificio de tipo investigador sobre qué tecnología debe desarrollar.

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades.

Postconcidiones:

- El edificio comienza la construcción.

6.3.19. Modelo de Comportamiento: Cancelar Desarrollo

Diagrama de comportamiento:

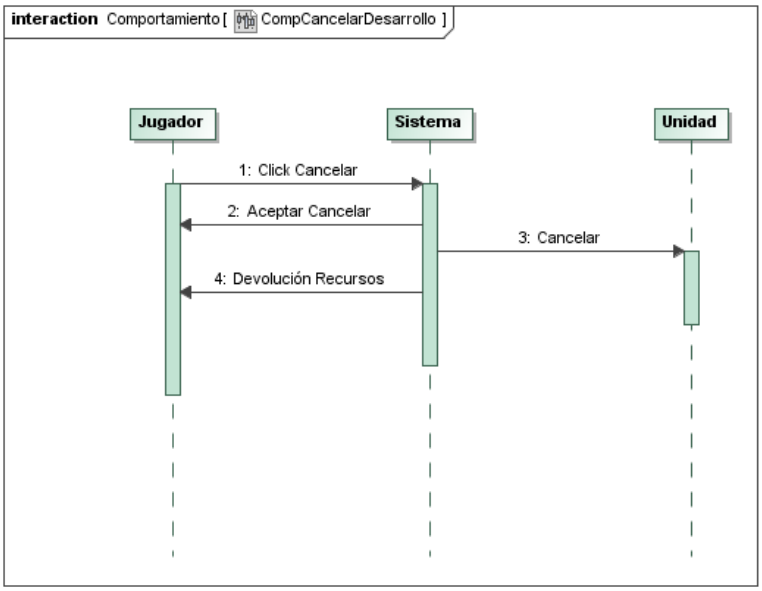

Figura 6.24: Diagrama Comportamiento: Crear Unidad.

Operación: Click Cancelar

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Cancelar Desarrollo".

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades o tecnologías.

Postconcidiones:

- El sistema inicia el principio de cambio de estado del edificio.

Operación: Aceptar Cancelar

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades o tecnologías.

Postconcidiones:

- El jugador vuelve a ganar el control.

Operación: Devolución Recursos

Responsabilidad: El sistema devuelve los recursos invertidos en la unidad / tecnología.

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades o tecnologías.

Postconcidiones:

- Los recursos del jugador se han incrementado.

Operación: Cancelar

Responsabilidad: El edificio deja de desarrollar su actividad actual.

Precondiciones: El jugador debe tener seleccionado un edificio propio que desarrolle unidades o tecnologías.

Postconcidiones:

- El desarrollo está cancelado y el edificio pasa a estado "Ocioso".

6.3.20. Modelo de Comportamiento: Comercio Recursos

Diagrama de comportamiento:

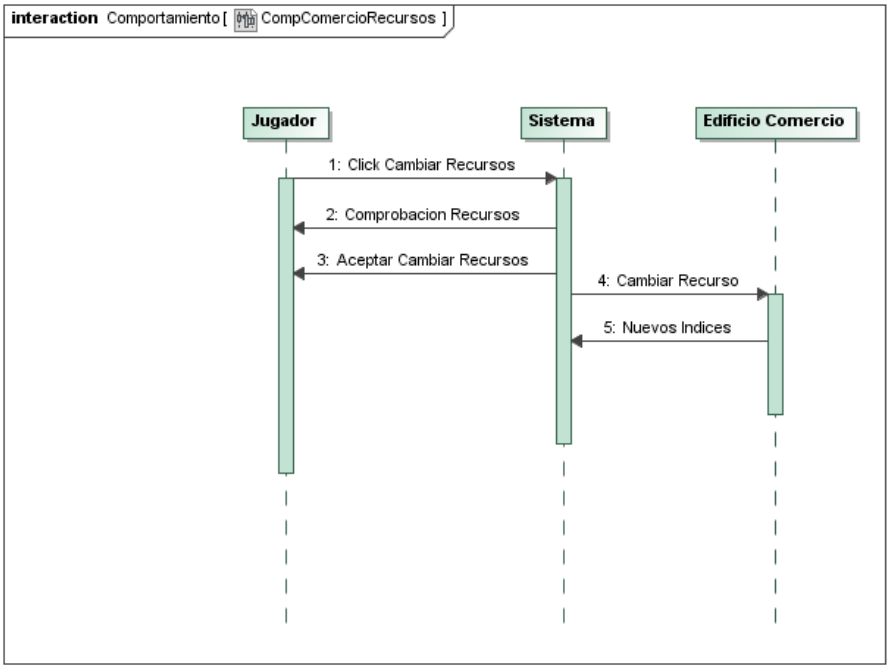

Figura 6.25: Diagrama Comportamiento: Crear Unidad.

Operación: Click Cambiar Recursos

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Cambiar Recursos".

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo comercio.

Postconcidiones:

- El sistema inicia el principio de cambio de estado del edificio.

Operación: Comprobación Recursos

Responsabilidad: El sistema comprueba que el usuario posee suficientes recursos para realizar el cambio.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo comercio.

Postconcidiones:

- El jugador es capaz de realizar el cambio.

Operación: Aceptar Cambiar Recursos

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo comercio.

Postconcidiones:

- El jugador vuelve a ganar el control.

Operación: Cambiar Recursos

Responsabilidad: El sistema comunica al edificio qué recurso quiere cambiar.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo comercio.

Postconcidiones:

- El edificio realiza el comercio y modifica los recursos de su propietario (jugador).

Operación: Nuevos Índices

Responsabilidad: El edificio de comercio actualiza los índices de cambio de mercado y se los manda al sistema.

Precondiciones: El jugador debe tener seleccionado un edificio propio de tipo comercio.

Postconcidiones:

- El sistema posee nuevos índices de intercambio de recursos.

6.3.21. Modelo de Comportamiento: Construir Edificio

Diagrama de comportamiento:

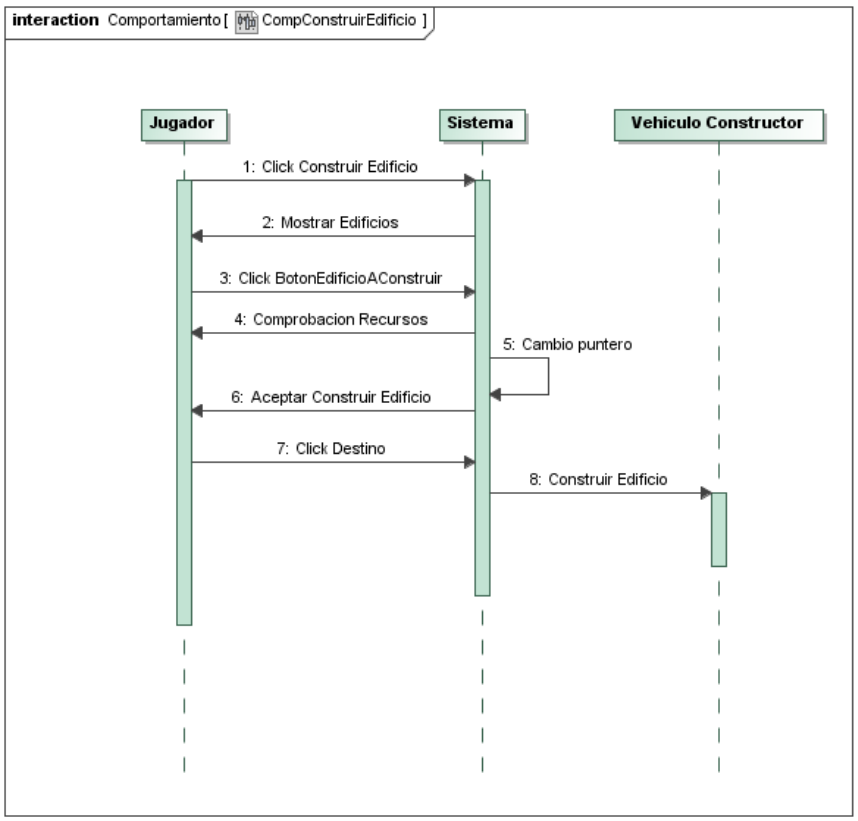

Figura 6.26: Diagrama Comportamiento: Construir Edificio.

Operación: Click Construir Edificio

Responsabilidad: Indica al sistema que la acción que se quiere realizar es "Construir Edificio".

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El sistema inicia el principio de cambio de estado de la unidad.

Operación: Mostrar Edificios

Responsabilidad: El sistema muestra los edificios disponibles a construir.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El jugador puede elegir qué edificio quiere construir.

Operación: Click BotonEdificioAConstruir

Responsabilidad: El jugador elige qué edificio quiere construir.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El sistema conoce el edificio a construir.

Operación: Comprobación Recursos

Responsabilidad: El sistema comprueba que el usuario posee suficientes recursos para realizar el cambio.

Precondiciones: El jugador debe tener seleccionado una unidad propia de tipo constructor.

Postconcidiones:

- El jugador es capaz de construir el edificio.

Operación: Cambio Puntero

Responsabilidad: Cambia la forma del puntero para que el usuario sepa que la acción ha sido aceptada.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo transporte.

Postconcidiones:

- El sistema cambia el puntero del ratón y muestra la sombra de construcción.

Operación: Aceptar Construir Edificio

Responsabilidad: El sistema devuelve el control al usuario.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El jugador vuelve a ganar el control.

Operación: Click Destino

Responsabilidad: El jugador indica la zona de construcción.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El sistema comprueba que la zona de construcción es viable y comienza a calcular el camino.

Operación: Construir Edificio

Responsabilidad: El sistema indica al vehículo qué edificio debe construir y dónde.

Precondiciones: El jugador debe tener seleccionado una unidad propia tipo constructor.

Postconcidiones:

- El vehículo cambia su estado a "Construir" y el vehículo empieza a moverse.

6.3.22. Modelo de Comportamiento: Comercio Recursos

Diagrama de comportamiento:

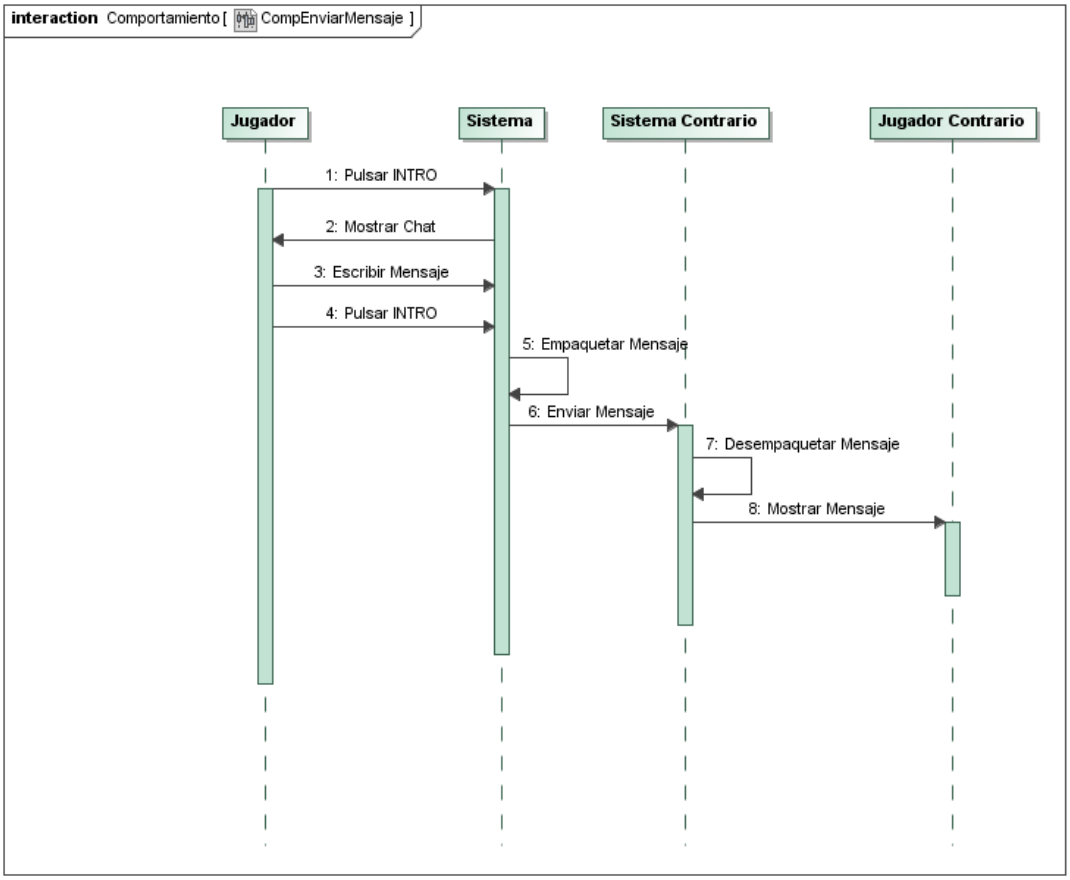

Figura 6.27: Diagrama Comportamiento: Enviar Mensaje.

Operación: Pulsar INTRO

Responsabilidad: El jugador indica al sistema que la acción que se quiere realizar es enviar un mensaje.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El sistema inicia el principio carga en memoria las estructuras para mostrar el chat.

Operación: Mostrar Chat

Responsabilidad: El sistema muestra una ventana para que el usuario escriba su mensaje.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El jugador termina de escribir su mensaje.

Operación: Pulsar INTRO

Responsabilidad: El jugador indica al sistema que ha terminado de escribir su mensaje.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El sistema deja de mostrar el chat.

Operación: Emapquetar Mensaje

Responsabilidad: El sistema prepara el mensaje para enviarlo al jugador contrario.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El sistema prepara una estructura para enviar el mensaje al jugador enemigo.

Operación: Enviar mensaje

Responsabilidad: El sistema envía el mensaje al sistema del jugador enemigo.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El sistema del jugador contrario posee el mensaje que el jugador ha escrito.

Operación: Desempaquetar Mensaje

Responsabilidad: El sistema contrario desempaqueta el mensaje para poder mostrárselo al jugador contrario.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El sistema contrario posee un mensaje en claro para mostrar.

Operación: Mostrar Mensaje

Responsabilidad: El sistema contrario muestra el mensaje por la pantalla del jugador contrario.

Precondiciones: El jugador debe estar en una partida multijugador.

Postconcidiones:

- El jugador contrario lee el mensaje.

Capítulo 7

Diseño del Sistema

Durante el proceso de análisis del sistema hemos obtenido la interacción y funciones del sistema, el modelo conceptual, las respuestas del sistema y operaciones del mismo. Durante el diseño obtendremos la arquitectura, clases software y operaciones.

El diseño del software forma parte del núcleo técnico del proceso de ingeniería del software y se aplica independientemente del paradigma de desarrollo utilizado. A partir del análisis y especificación de los requisitos del software, el diseño es la primera de las tres actividades técnicas (diseño, codificación y prueba) que hay que realizar para construir y verificar el software. Es un proceso iterativo a través del cual se traducen los requisitos en un modelo o representación del sistema que se va a construir. Inicialmente, el diseño se representa a un alto nivel de abstracción y a medida que se realizan iteraciones, el refinamiento siguiente lleva a representaciones del diseño aun nivel mucho más detallado.

7.1. Arquitectura del sistema software

En el apartado presente daremos una descripción de los subsistemas existentes y de los componentes con sus relaciones del sistema software. Dentro de la determinación de la arquitectura software necesitamos establecer que propiedades ha de cumplir el sistema y de que recursos disponemos.

La arquitectura del software es la estructura del sistema, componentes, propiedades y relaciones.

En nuestra aplicación podemos encontrar una división en capas de la arquitectura.

- Capa de interfaz de usuario: Incluye todos los componentes utilizados para representar la interfaz de usuario de la aplicación.
- Capa de control de aplicación: Incluye componentes y procedimientos que permiten el control de eventos y mandar mensajes a las capas necesarias.
- Capa de dominio: Incluye todas las clases con la que trabaja el sistema.

Como hemos comentado anteriormente nuestro sistema utiliza lenguaje C++ que posee la capacidad de trabajar con orientación a objetos.

7.2. Diagramas de secuencia

Mediante los diagramas de secuencia definiremos la interacción entre las clases de objetos en respuesta a los eventos producidos en el sistema. Utilizaremos los diagramas de comportamiento ya que éstos dan a conocer de una forma efectiva el orden de los mensajes entre objetos y eventos. Utilizaremos cajas para representar objetos, clases y multiobjetos. Los mensajes seguirán la siguiente estructura.

7.2.1. Un Jugador

Diagrama de secuencia:

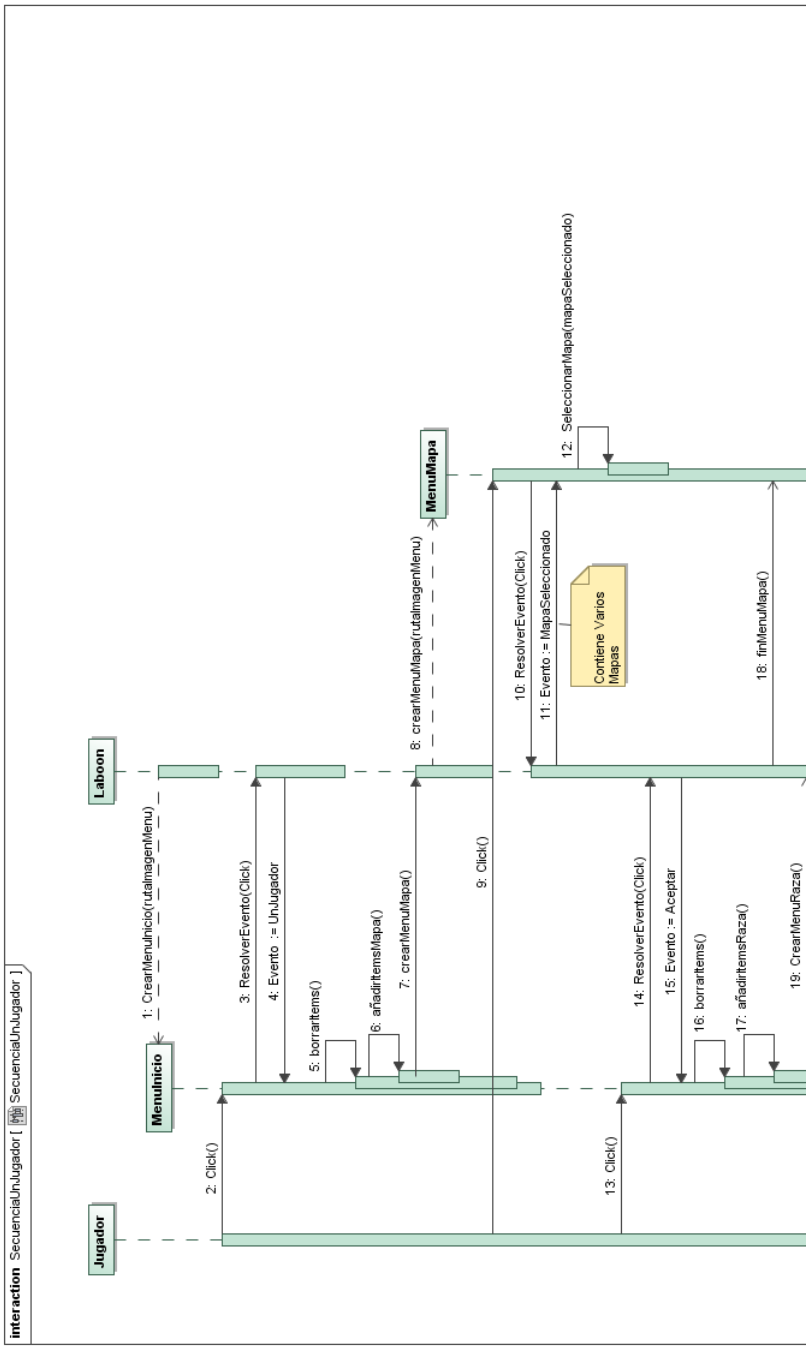

Figura 7.1: Diagrama Secuencia: Un Jugador (Parte 1).

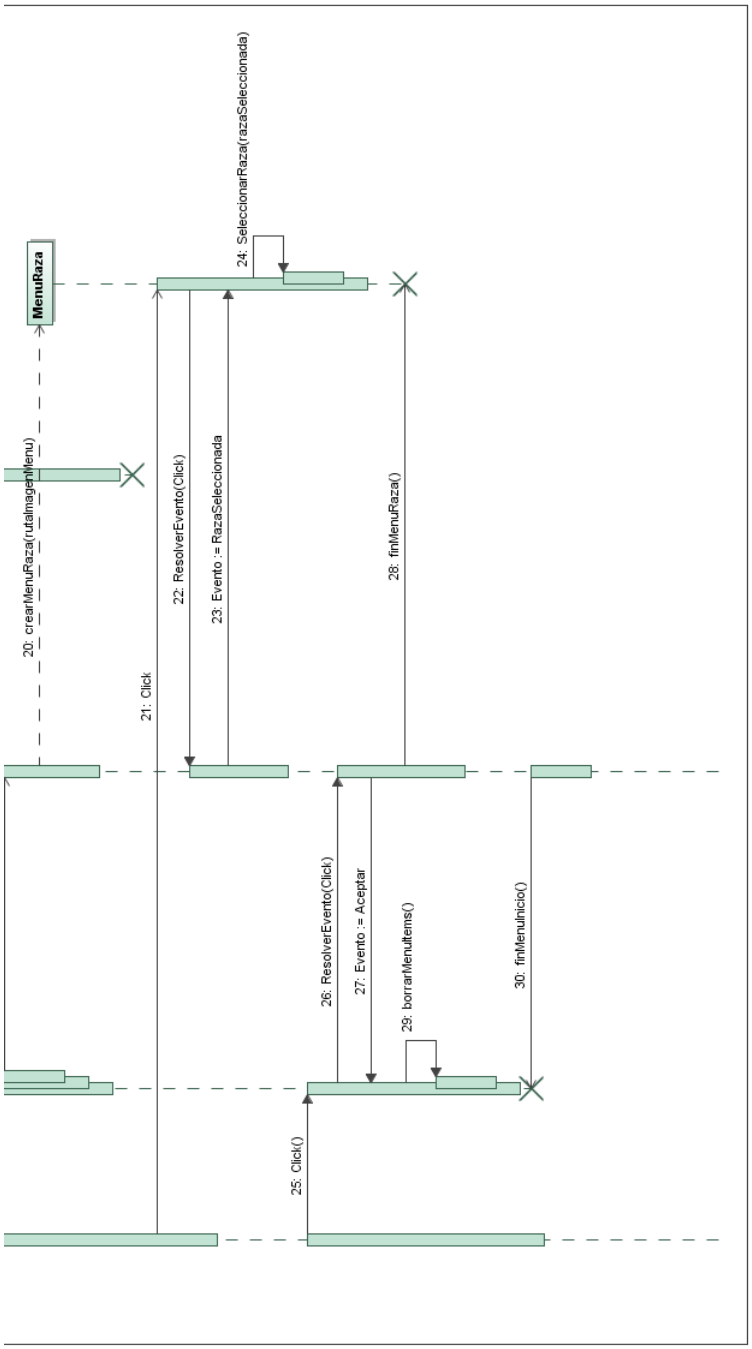

Figura 7.2: Diagrama Secuencia: Un Jugador (Parte 2).

El jugador al arrancar la aplicación se encontrará con la el menú principal de *MenuInicio*. Al seleccionar "Un Jugador" el sistema le redirigirá automáticamente a la parte del menú de la selección de mapas. El jugador elegirá el mapa que desee entre la lista proporcionada y pulsará aceptar. Tras esto se guardarán los datos del mapa seleccionado y se destruirá *MenuMapa* y el jugador se encontrará con el menú de selección de bando. Éste elegirá uno de las tres bandos disponibles y pulsará aceptar. El sistema almacenará los datos del bando seleccionado, eliminará *MenuRaza* y *MenuInicio* y dará comienzo la partida.

7.2.2. Crear Partida Multijugador

Diagrama de secuencia:

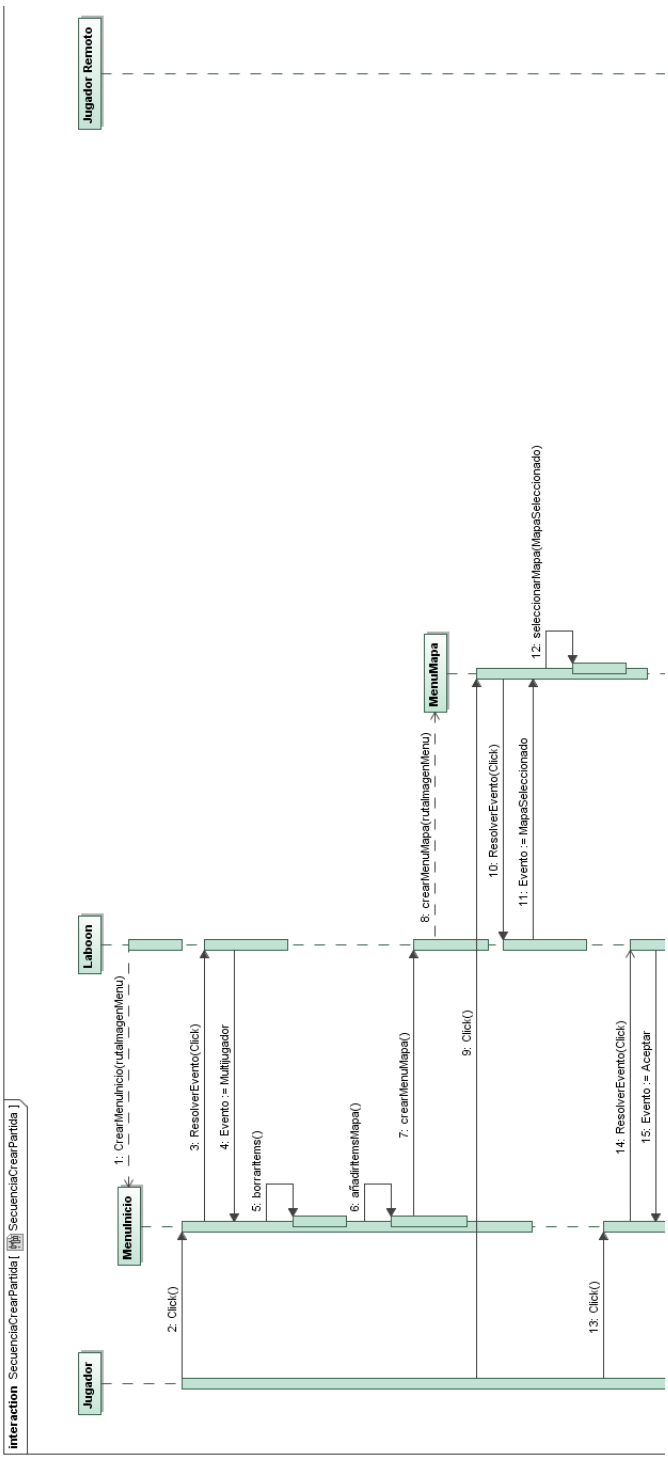

Figura 7.3: Diagrama Secuencia: Crear Partida (Parte 1).

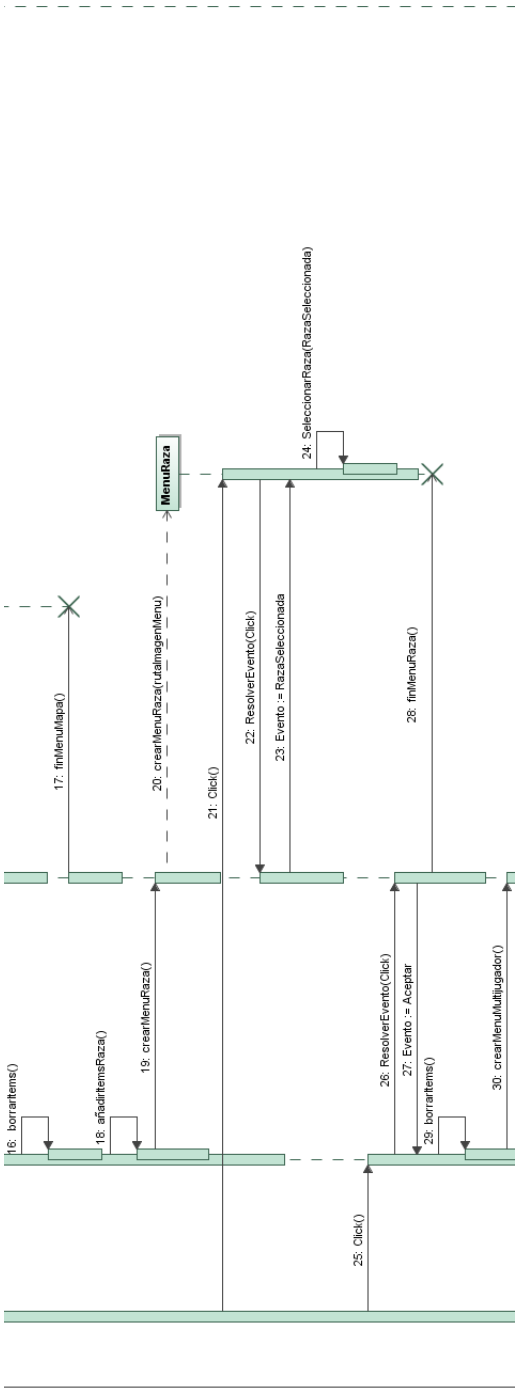

Figura 7.4: Diagrama Secuencia: Crear Partida (Parte 2).

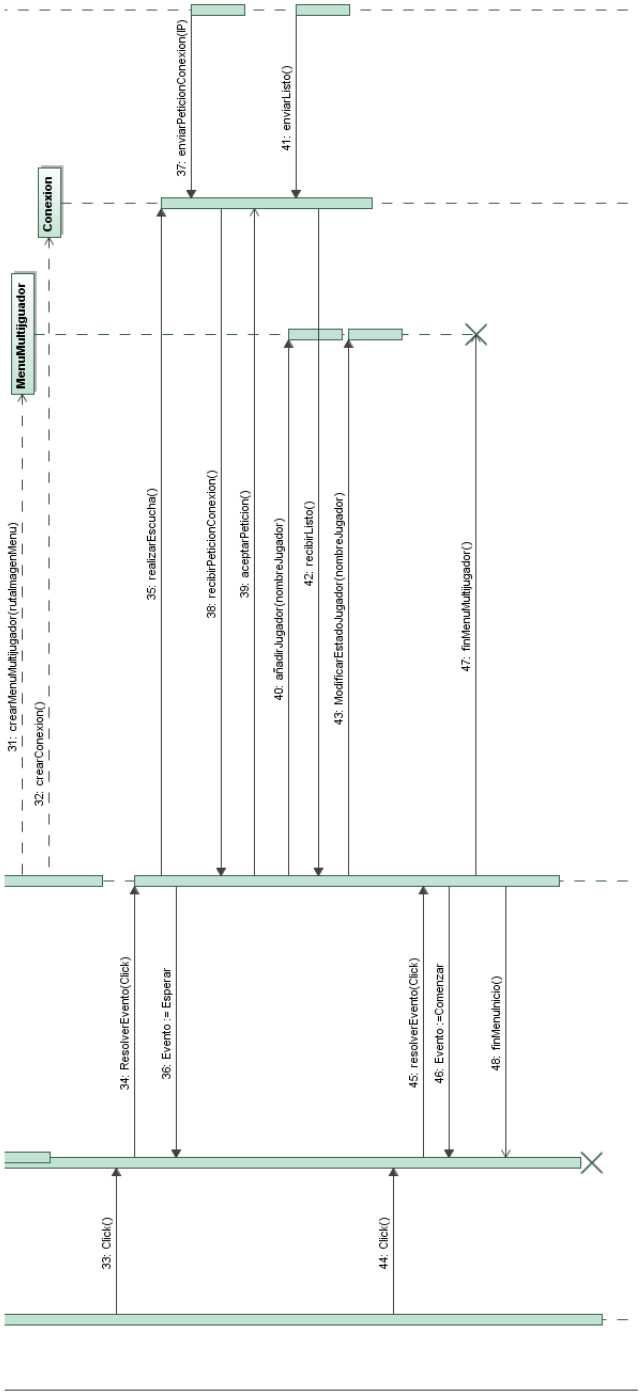

Figura 7.5: Diagrama Secuencia: Crear Partida (Parte 3).

El jugador decide crear una partida multijugador, para ello seleccionará "Multijugador" en el *MenuInicio* y luego pulsará "Crear Partida" redirigiendo al jugador al formulario de introducción de nombre, en el cual introducirá su nombre y luego pulsará la tecla "Intro" y pulsará aceptar. Esto lo llevará al menú de selección de mapas. El jugador eligirá el mapa que desea entre la lista de mapas y pulsará aceptar. Tras guardar los datos del mapa seleccionado, el sistema destruirá *MenuMapa* y el jugador se encontrará en el menú de selección de bando. Éste elegirá una de entre los tres bandos disponibles y pulsará aceptar. El sistema almacenará los datos del bando seleccionado y eliminará *MenuRaza* para dar paso a mostrar el *MenuPartidaMultijugador*. Cuando el jugador se encuentra en este menú pulsará el botón "Esperar" para recibir peticiones de conexión del jugador remoto. Una vez establecida dicha conexión, el jugador remoto avisará mediante un mensaje que está en estado "Listo" y el jugador podrá darle a "Empezar" eliminando *MenuPartidaMultijugador* y *MenuInicio* y dándole la misma orden de comenzar al jugador remoto.

7.2.3. Unirse Partida Multijugador

Diagrama de Secuencia:

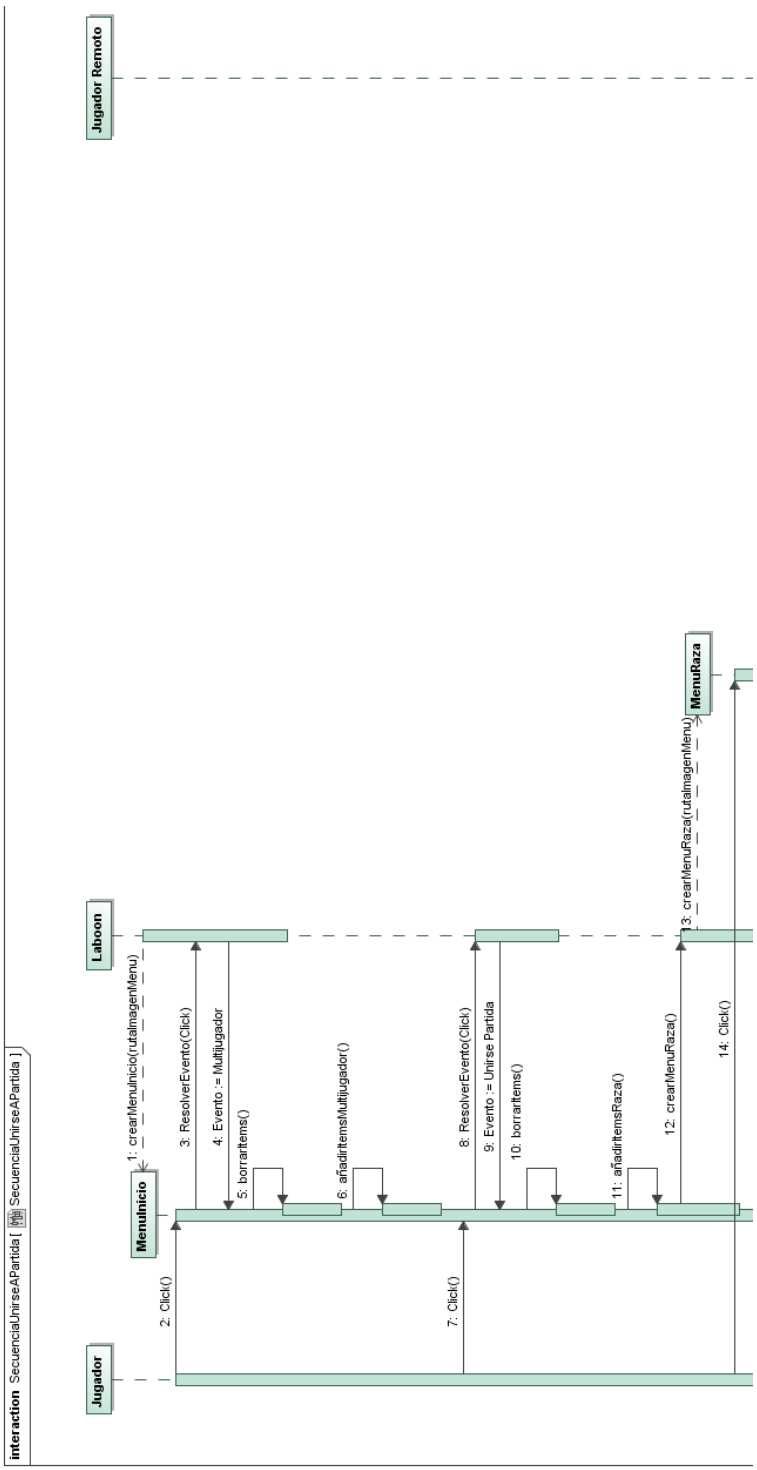

Figura 7.6: Diagrama Secuencia: Unirse A Partida (Parte 1).

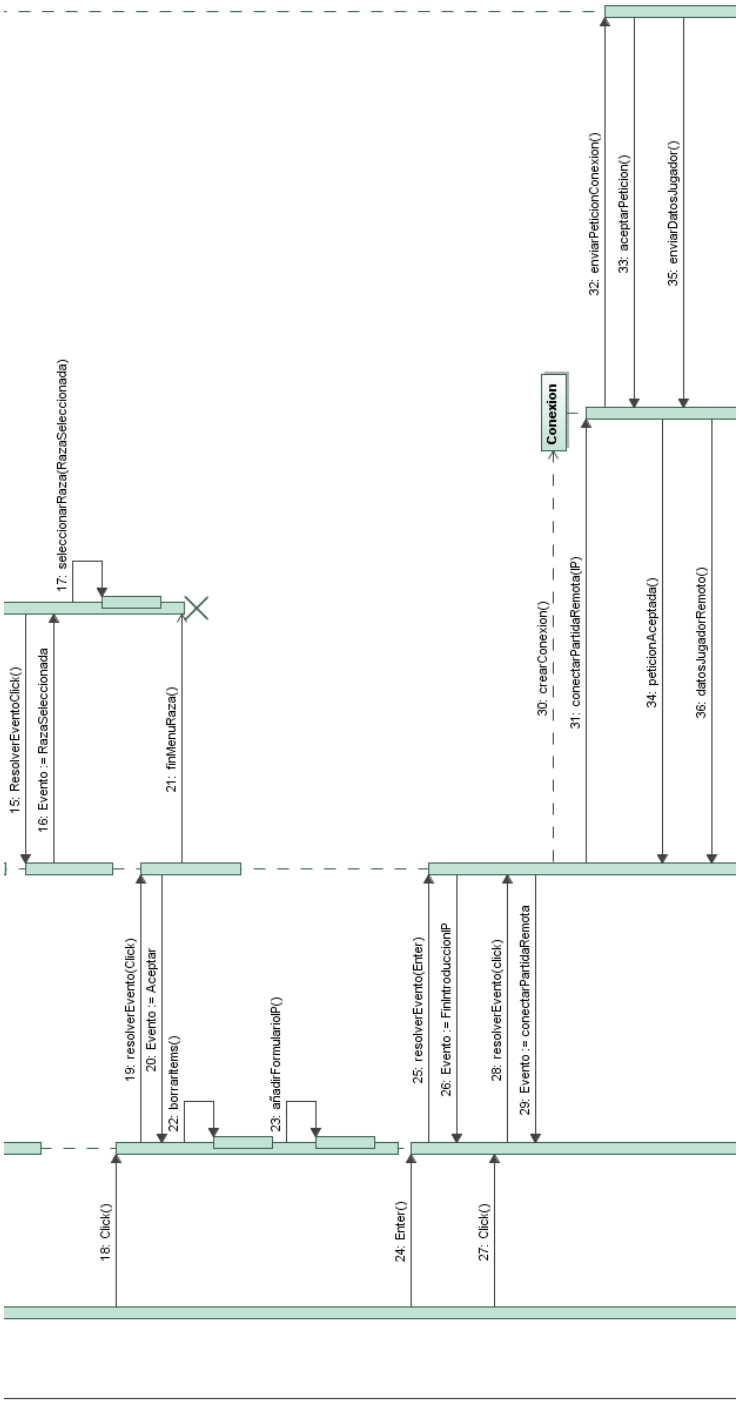

Figura 7.7: Diagrama Secuencia: Unirse A Partida (Parte 2).

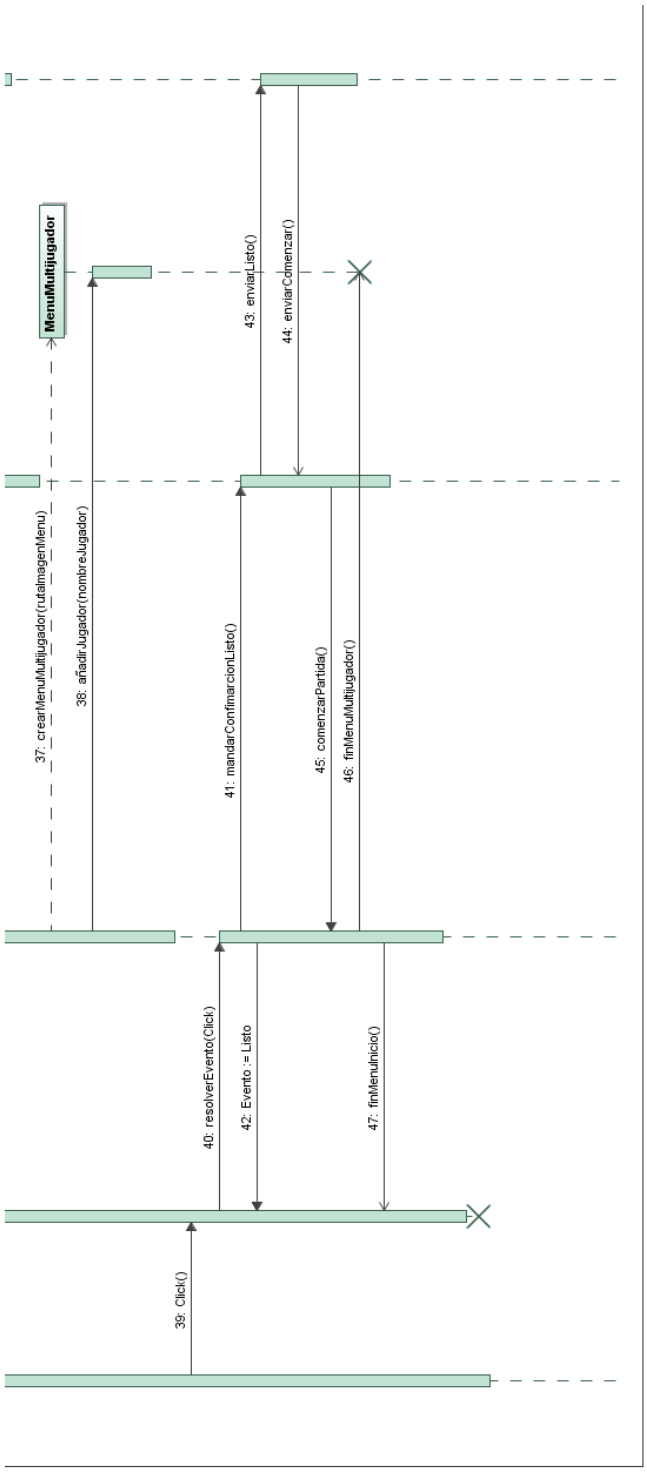

Figura 7.8: Diagrama Secuencia: Unirse A Partida (Parte 3).

El jugador decide unirse a una partida multijugador creada por un jugador remoto, para ello seleccionará "Multijugador" en el *MenuInicio* y luego pulsará "Unirse A Partida" redirigiendo al jugador al formulario de introducción de nombre, en el cual introducirá su nombre y luego pulsará la tecla "Intro" y pulsará el botón aceptar. Esto lo redirigirá al menú de selección de bando. Éste elegirá una de entre los tres bandos disponibles y pulsará aceptar. El sistema almacenará los datos del bando seleccionado y eliminará *MenuRaza* para dar paso al formulario de introducción de IP. El jugador introducirá la IP del equipo donde el jugador remoto tiene creada la partida, pulsará la tecla "Intro" y pulsará aceptar. Si la IP es correcta y el jugador remoto se encontraba en espera recibirá un mensaje para entrar en el *MenuPartidaMultijugador*. Cuando el jugador se encuentre listo le dará al botón "Listo" avisando al servidor de que se encuentra todo preparado para comenzar la partida y éste arrancará la partida, dando órdenes al jugador de que haga lo mismo eliminando *MenuPartidaMultijugador* y *MenuInicio*.

7.2.4. Cambiar Resolución

Diagrama de Secuencia:

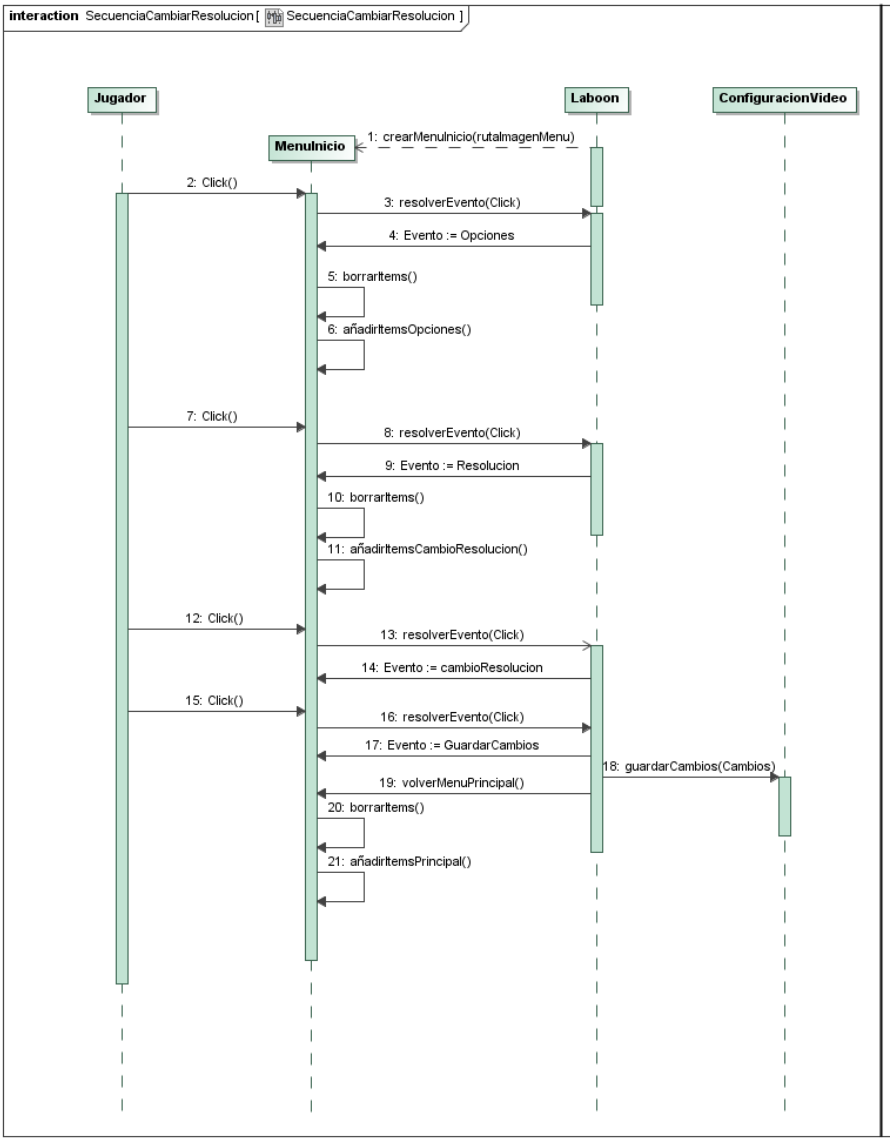

Figura 7.9: Diagrama Secuencia: Cambiar Resolución.

El jugador en este caso quiere cambiar una de las opciones de configuración del sistema, para ello seleccionará "Opciones" en el *MenuInicio* y luego pulsará la opción "Resolución". Esto provocará que el jugador sea redirigido a un menú donde podrá contemplar las distintas resoluciones que posee la aplicación y elegirá una de ellas. Tras esto pulsará en aceptar y provocará una cadena

de llamadas para que por una parte se guarden los cambios solicitados por el jugador y, por otra parte se actualice el estado actual de *Laboon*. Después el jugador es redirigido al menú principal.

7.2.5. Cambiar Aceleración

Diagrama de Secuencia:

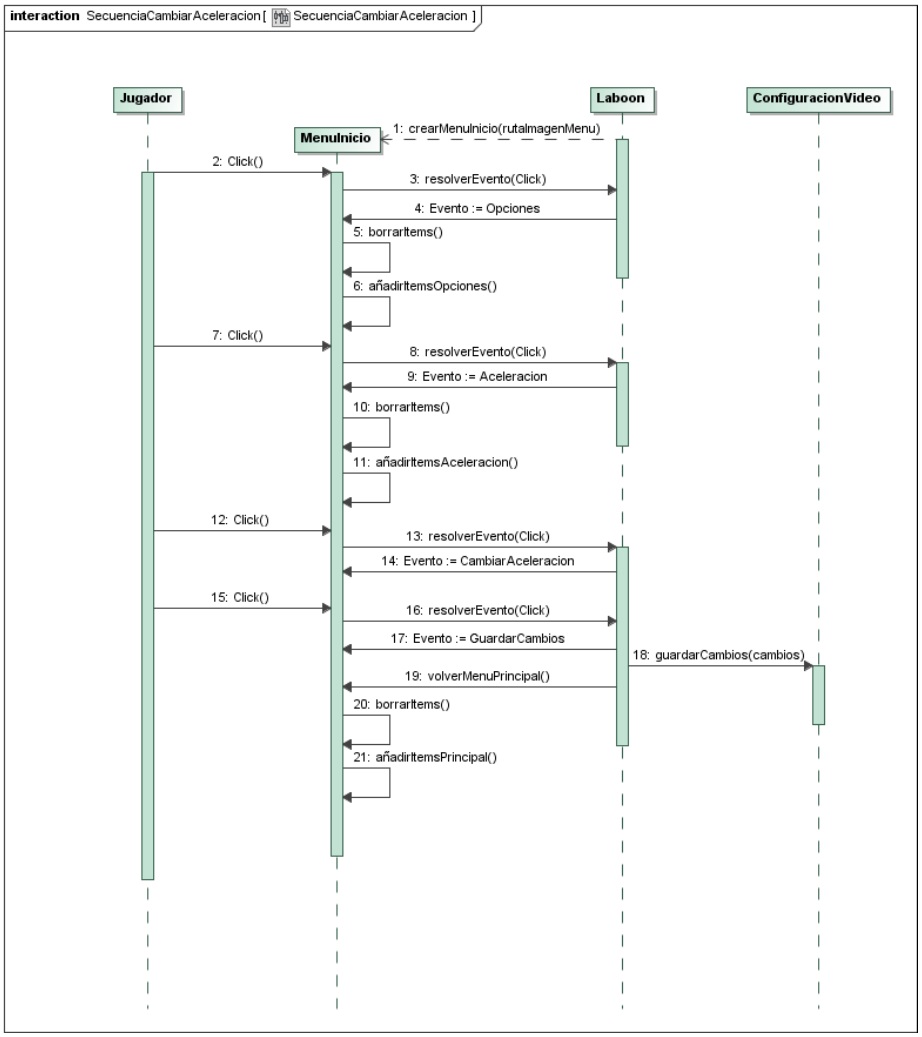

Figura 7.10: Diagrama Secuencia: Cambiar Aceleración.

El jugador en este caso quiere cambiar una de las opciones de configuración del sistema, para ello seleccionará "Opciones" en el *MenuInicio* y luego pulsará la opción "Aceleración". Esto provocará que el jugador sea redirigido a un menú donde podrá contemplar las distintas aceleraciones que posee la aplicación y elegirá una de ellas. Tras esto pulsará en aceptar y provocará una cadena de llamadas para que por una parte se guarden los cambios solicitados por el jugador y, por otra parte se actualice el estado actual de *Laboon*. Después el jugador es redirigido al menú principal.

7.2.6. Activar o Desactivar Sistema de Audio

Diagrama de Secuencia:

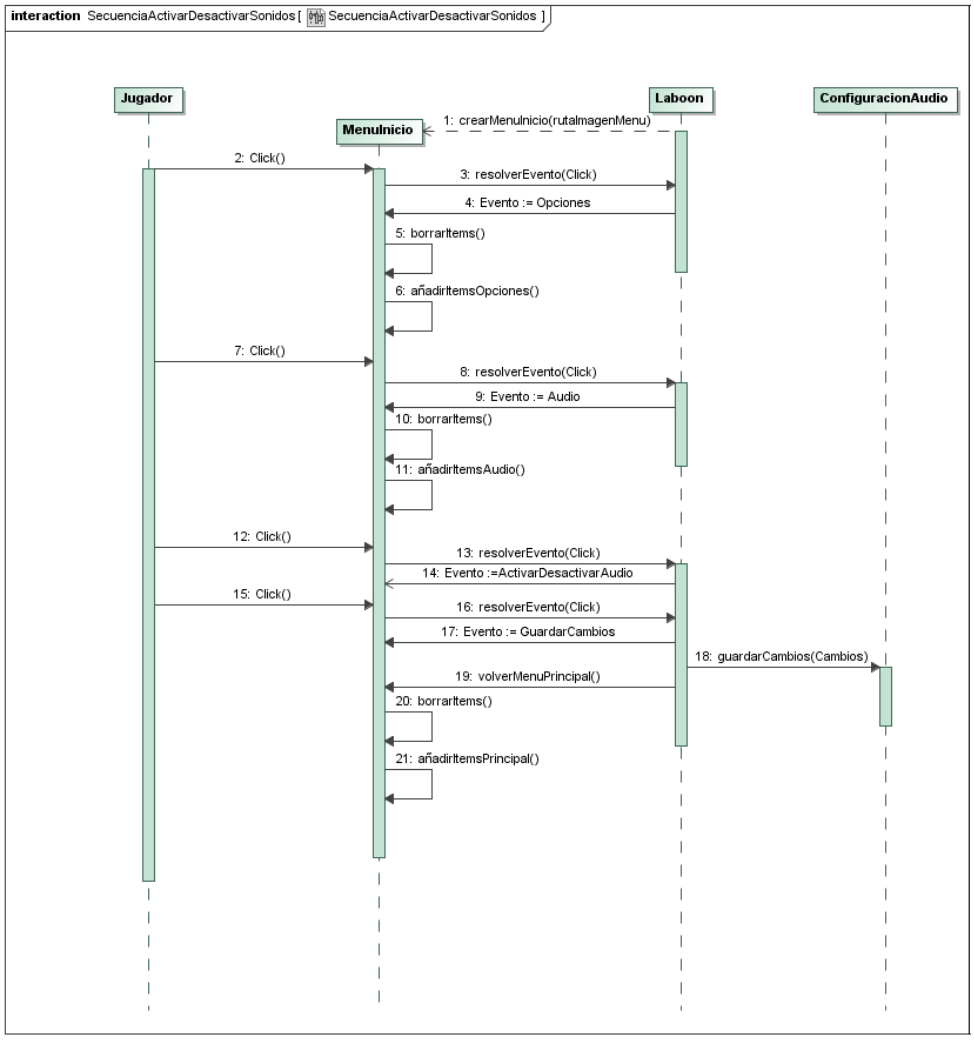

Figura 7.11: Diagrama Secuencia: Activar o Desactivar Sistema de Audio.

El jugador en este caso quiere cambiar una de las opciones de configuración del sistema, para ello seleccionará "Opciones" en el *MenuInicio* y luego pulsará la opción "Sonidos". Esto provocará que el jugador sea redirigido a un menú donde podrá contemplar el sistema de audio y elegirá si activarlo o desactivarlo. Tras esto pulsará en aceptar y provocará una cadena de llamadas para que por una parte se guarden los cambios solicitados por el jugador y, por otra parte se actualice el estado actual de *Laboon*. Después el jugador es redirigido al menú principal.

7.2.7. Activar o Desactivar Sistema de Subtítulos

Diagrama de Secuencia:

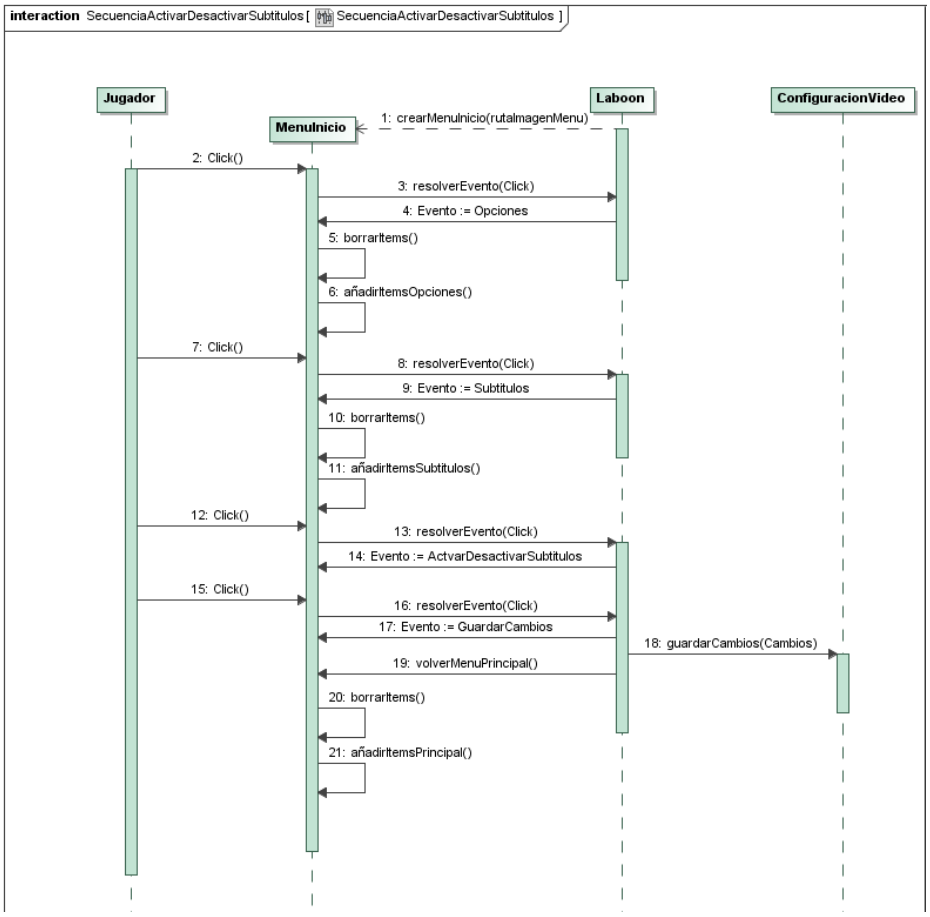

Figura 7.12: Diagrama Secuencia: Activar o Desactivar Sistema de Subtítulos.

El jugador en este caso quiere cambiar una de las opciones de configuración del sistema, para ello seleccionará "Opciones" en el *MenuInicio* y luego pulsará la opción "Sonidos". Esto provocará que el jugador sea redirigido a un menú donde podrá contemplar el sistema de audio y elegirá si activarlo o desactivarlo. Tras esto pulsará en aceptar y provocará una cadena de llamadas para que por una parte se guarden los cambios solicitados por el jugador y, por otra parte se actualice el estado actual de *Laboon*. Después el jugador es redirigido al menú principal.

7.2.8. Mover Unidad

Diagrama de Secuencia:

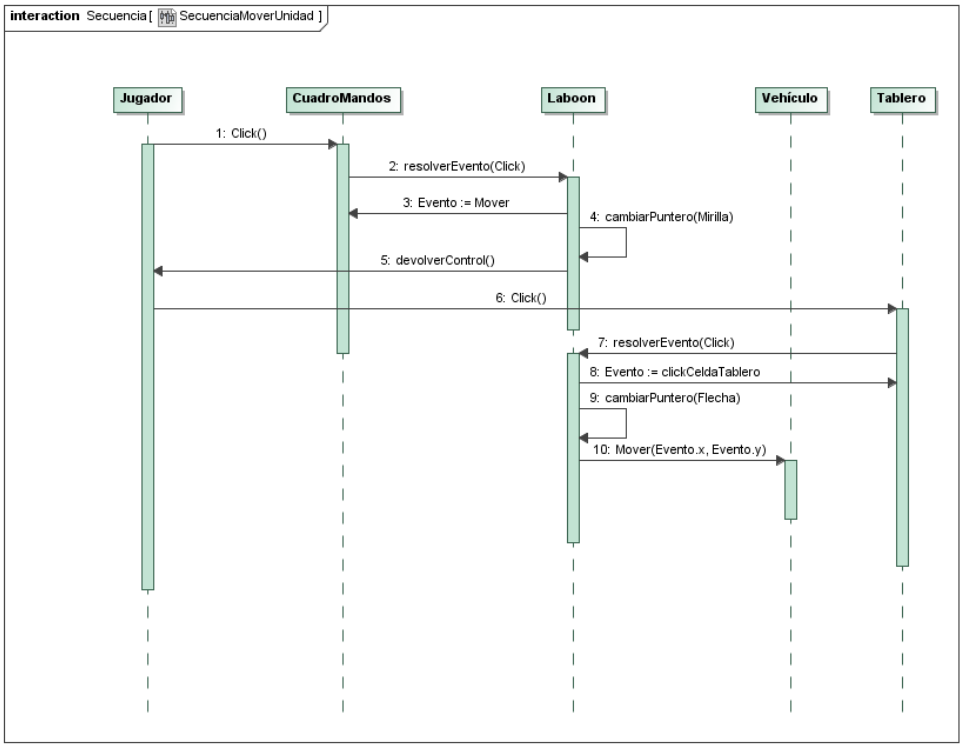

Figura 7.13: Diagrama Secuencia: Mover Unidad.

El jugador quiere mover una o varias unidades propias que tiene seleccionadas hasta un punto destino, hará click en el botón "mover" inicialmente y seguidamente hará click en la zona del mapa donde quiera que todas las unidades se desplacen.

7.2.9. Patrullar Unidad

Diagrama de Secuencia:

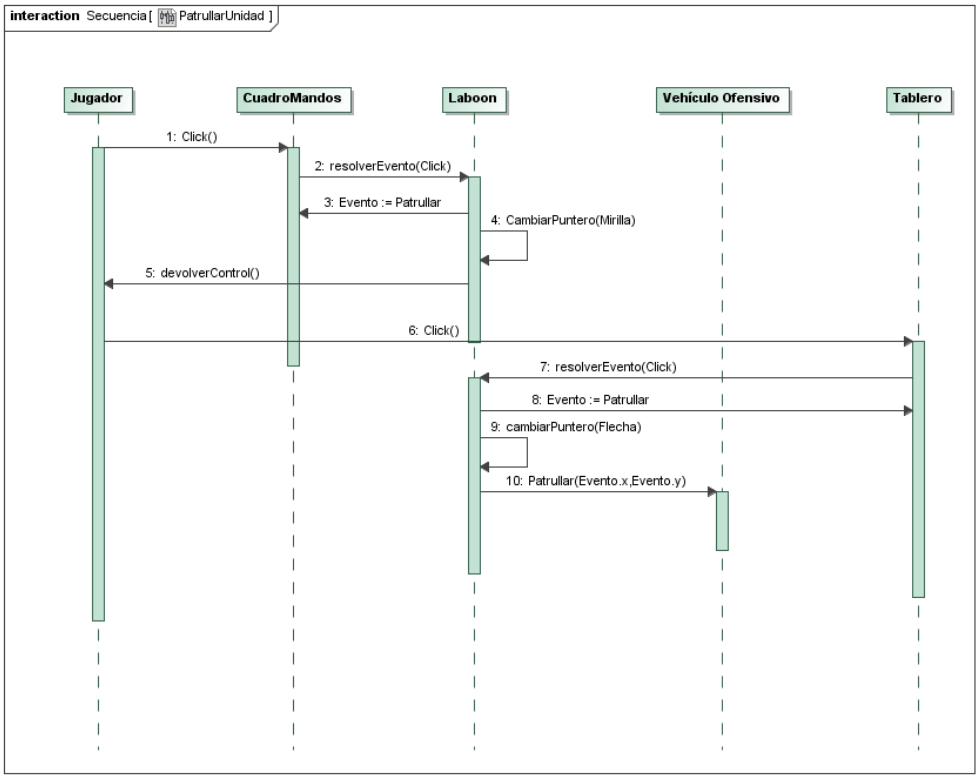

Figura 7.14: Diagrama Secuencia: Patrullar Unidad.

El jugador quiere que una unidad ofensiva patrulle desde su posición actual hasta un punto destino que seleccione, para ello seleccionará el botón “patrullar” y seguidamente hará click en la zona del mapa donde quiera que dicha unidad haga la patrulla

7.2.10. Parar Unidad

Diagrama de Secuencia:

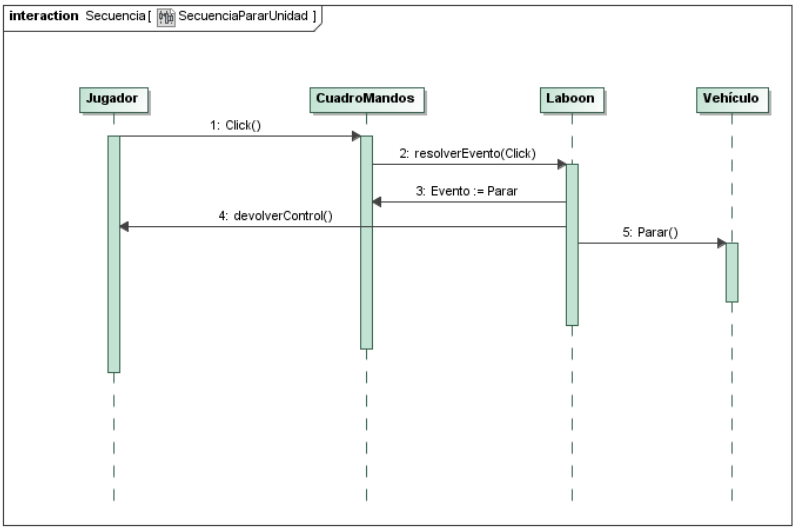

Figura 7.15: Diagrama Secuencia: Parar Unidad.

El jugador quiere que una o varias unidades dejen de hacer lo que estén haciendo y se paren, para ello seleccionará el botón “parar”.

7.2.11. Atacar Unidad

Diagrama de Secuencia:

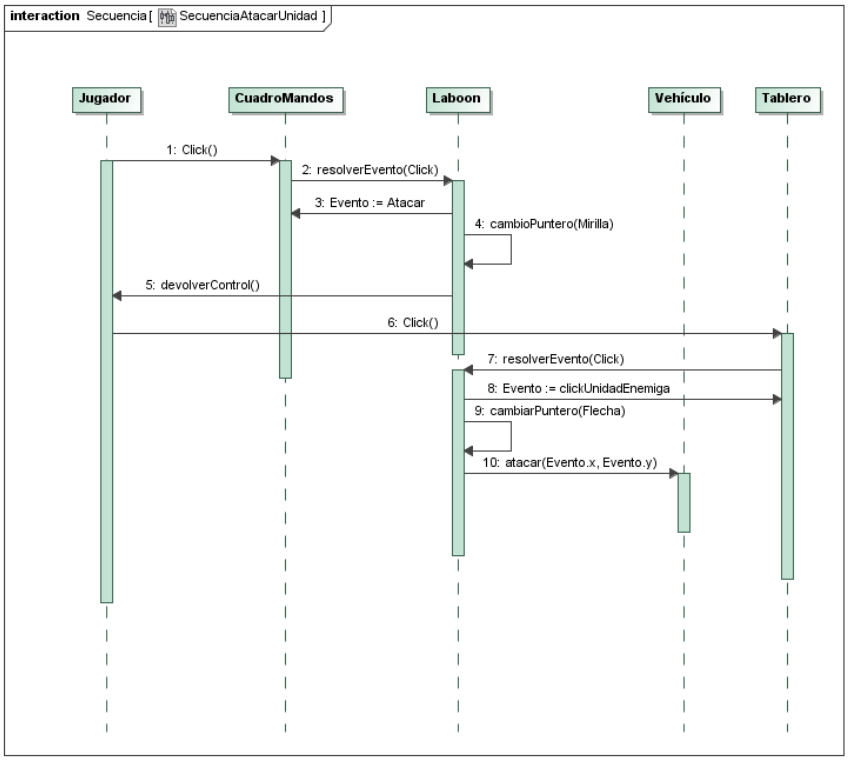

Figura 7.16: Diagrama Secuencia: Atacar Unidad.

El jugador quiere mover una o varias unidades propias que tiene seleccionadas hasta un punto destino, hará click en el botón “atacar” inicialmente y seguidamente hará click en la zona del mapa donde se encuentre un enemigo al que atacar. Si no hay ningún enemigo en la zona seleccionada, se desplazarán hasta la zona objetivo.

7.2.12. Mantener Posición Unidad

Diagrama de Secuencia:

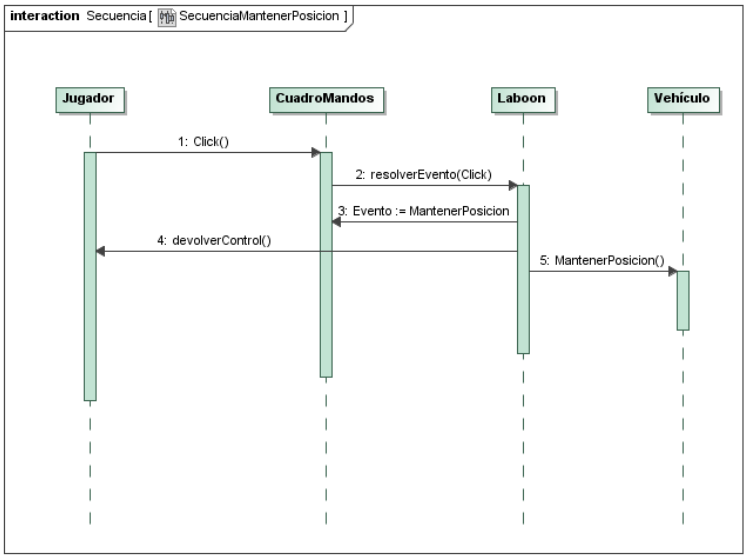

Figura 7.17: Diagrama Secuencia: Mantener Posición Unidad.

El jugador quiere que una o varias unidades propias que tiene seleccionadas mantengan su posición actual a toda costa, para ello hará click en el botón "mantener poisicion" para que todas las unidades dejen de moverse y estén conservando su posición actual.

7.2.13. Reparar Edificio

Diagrama de Secuencia:

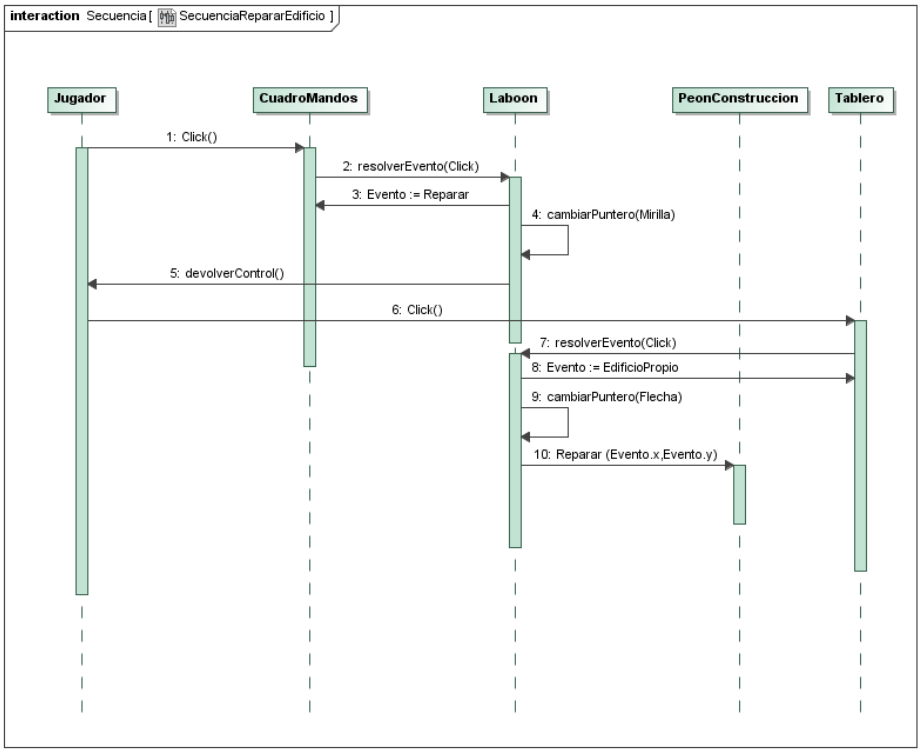

Figura 7.18: Diagrama Secuencia: Reparar Edificio.

El jugador quiere que reparar uno de sus edificio, para ello seleccionará una unidad de tipo “PeonConstruccion” y le dará al botón “Reparar” en el cuadro de mandos, tras esto seleccionará en el tablero que edificio quiere reparar. Si no existe edificio en la zona que quiere reparar se desplazará a la zona indicada.

7.2.14. Recolectar Recursos

Diagrama de Secuencia:

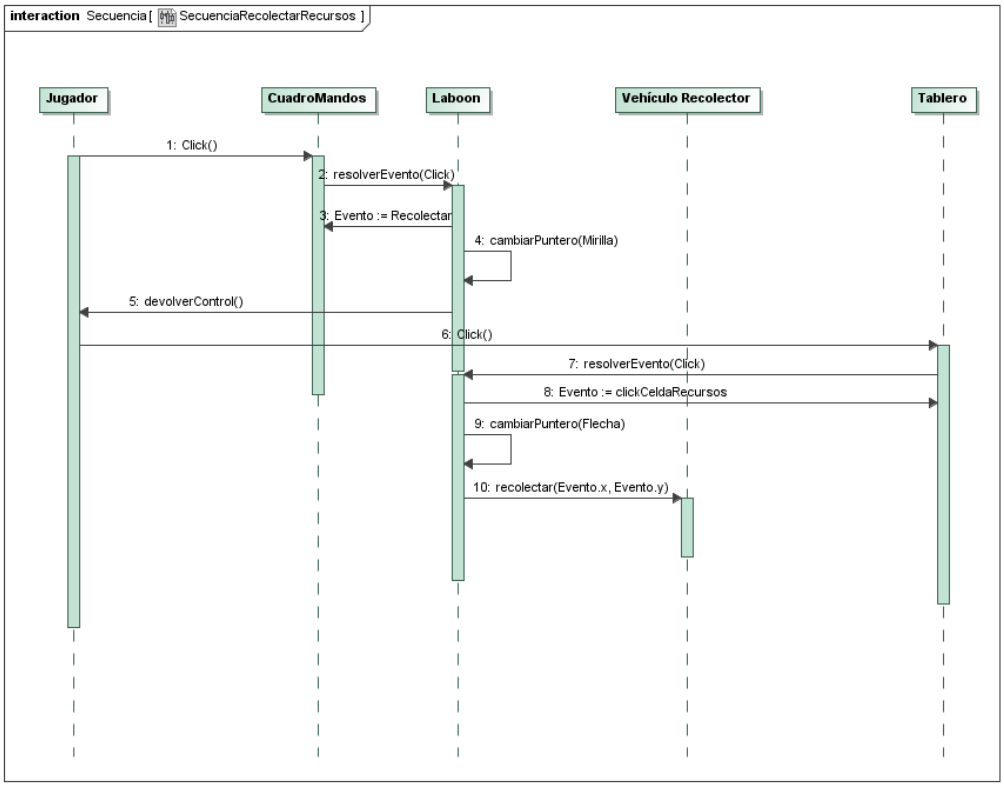

Figura 7.19: Diagrama Secuencia: Recolectar Recursos.

El jugador quiere que recolectar recursos de una zona específica del mapa, por tanto seleccionará a un "PeonLeñador" si quiere recolectar madera, a un "PeonExcavador" si quiere recolectar metal terrestre o a un "BarcoPeon" si quiere recolectar metal submarino. Seleccionará el botón "Recolectar" y pulsará seguidamente en la zona del mapa donde quiera recolectar. Si no hay recursos en la zona especificada la unidad se desplazará hasta la zona indicada.

7.2.15. Cargar Unidad

Diagrama de Secuencia:

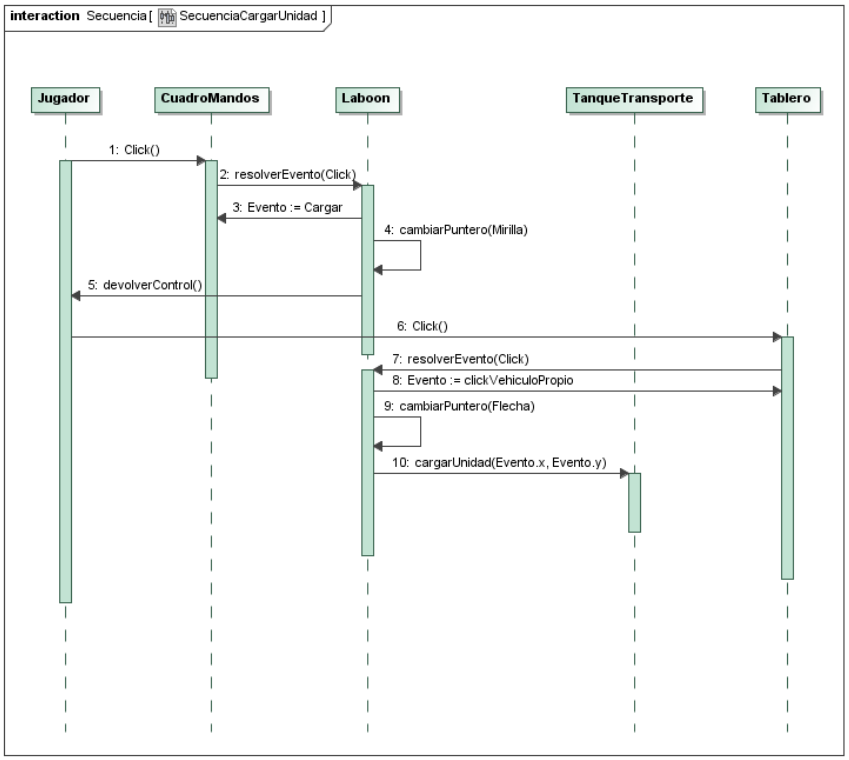

Figura 7.20: Diagrama Secuencia: Cargar Unidad.

El jugador quiere cargar una unidad dentro de uno de sus transportes, para ello seleccionará una unidad de tipo “TanqueTransporte” y le dará al botón “Cargar”, tras esto seleccionará la unidad que desea cargar, si no existe unidad en la zona especificada, el transporte se desplazará hasta allí.

7.2.16. Descargar Unidades

Diagrama de Secuencia:

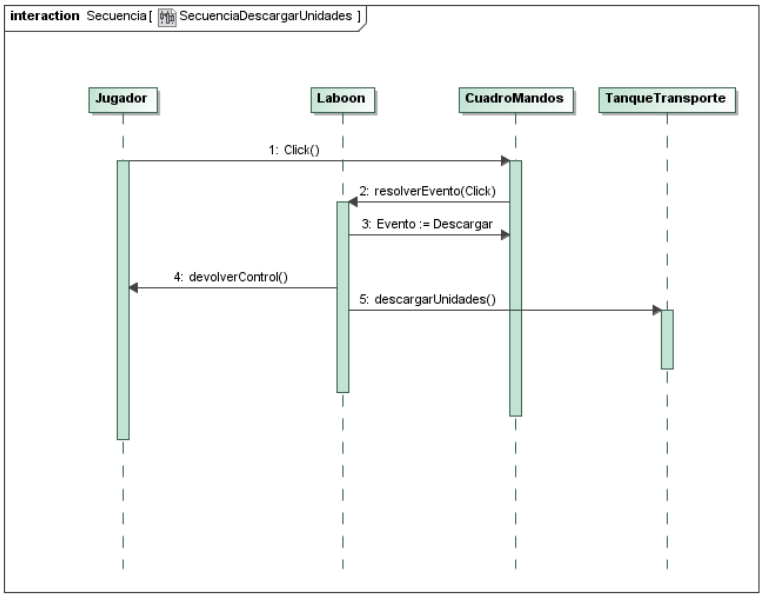

Figura 7.21: Diagrama Secuencia: Descargar Unidades.

El jugador quiere descargar todas las unidades que posee uno de sus transportes, para ello seleccionará una unidad de tipo “TanqueTransporte” y le dará al botón “Descargar”, la unidad de transporte dejará de hacer lo que esté haciendo para pararse, y descargar todas las unidades que estén dentro, si existe espacio suficiente alrededor.

7.2.17. Construir Edificio

Diagrama de Secuencia:

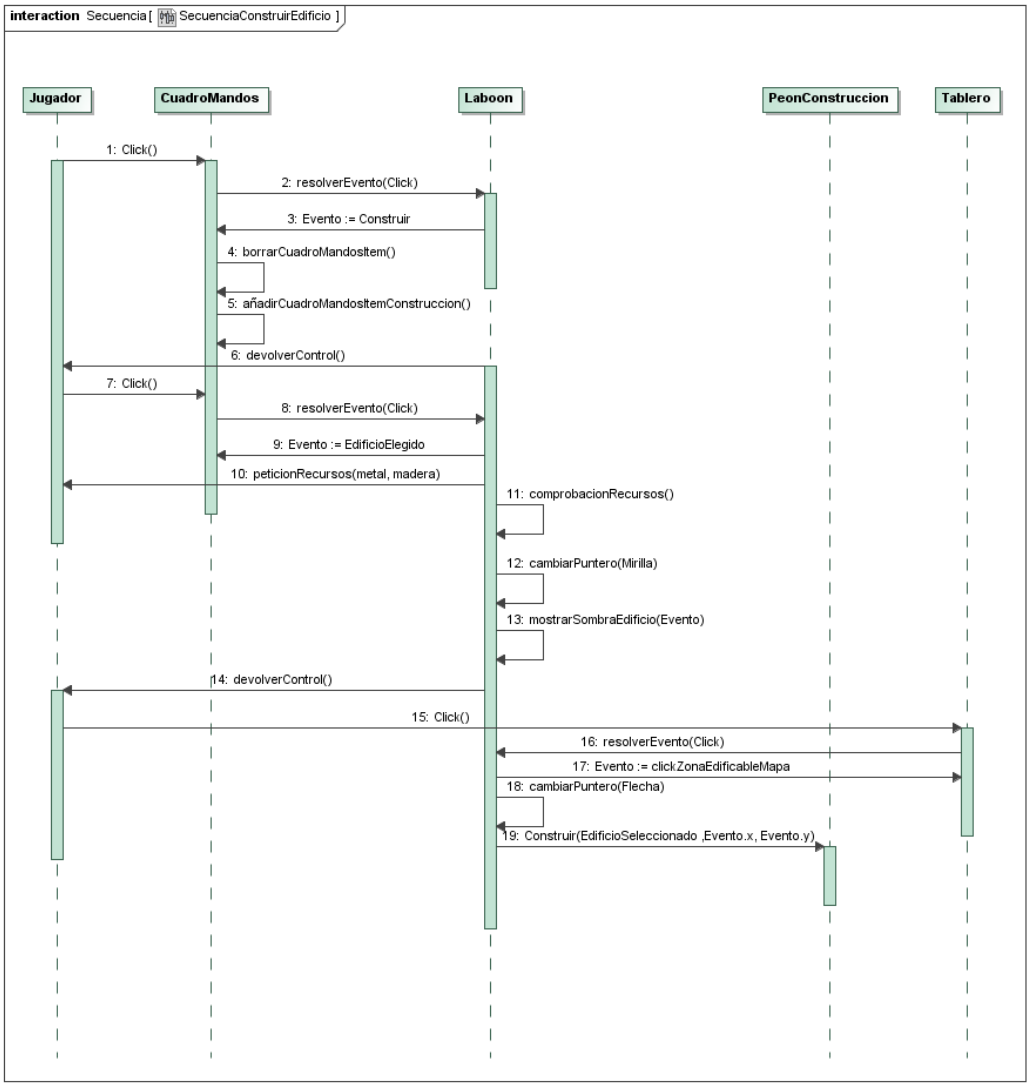

Figura 7.22: Diagrama Secuencia: Construir Edificio.

El jugador quiere construir un edificio, para ello seleccionará una unidad de tipo "PeonConstruccion" y seleccionará "Construir" tras esto en el cuadro de mandos aparecerán todos los edificios disponibles para construir. Tras esto seleccionará el edificio que quiere construir y si tiene recursos suficientes aparecerá en la posición del cursor la sombra de construcción del edificio y elegirá donde quiere construirlo.

7.2.18. Desarrollar Tecnología

Diagrama de Secuencia:

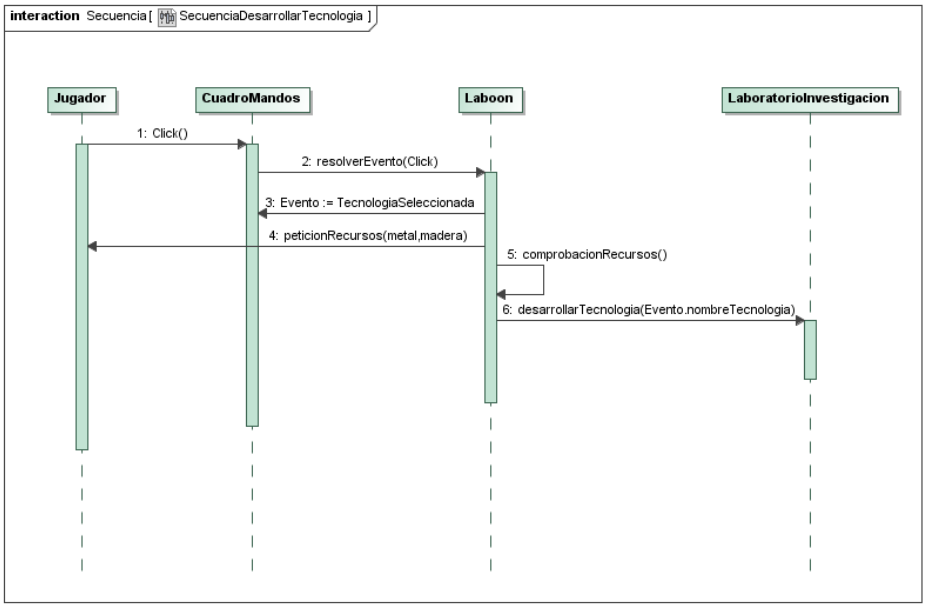

Figura 7.23: Diagrama Secuencia: Desarrollar Tecnología.

El jugador quiere mejorar sus unidades y/o edificios a través de la mejora tecnológica del juego. Para ello tendrá que seleccionar un edificio del tipo "LaboratorioInvestigacion" y luego hacer click sobre cualquiera de las tecnologías que quiera mejorar, el sistema comprobará si tiene recursos suficientes y procederá a realizar la mejora.

7.2.19. Crear Unidad

Diagrama de Secuencia:

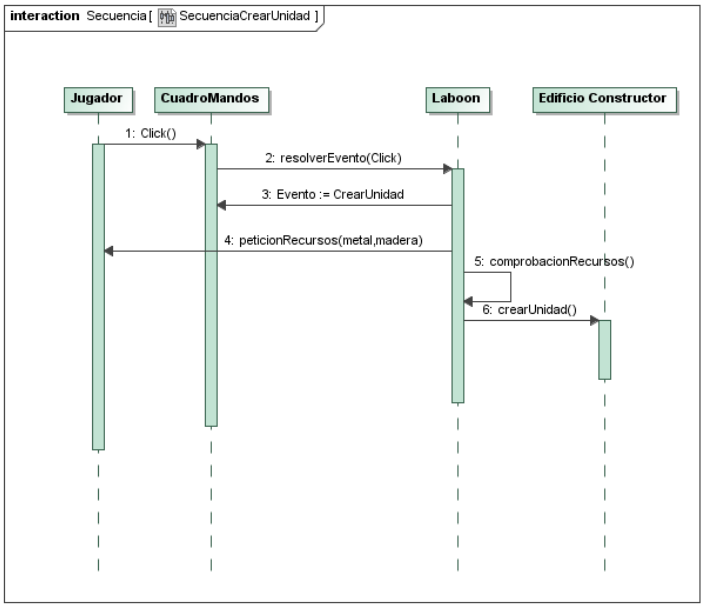

Figura 7.24: Diagrama Secuencia: Crear Unidad.

El jugador quiere crear un vehículo nuevo, para ello debe seleccionar cualquiera de los edificios creadores de unidades, según lo que desee crear y picar sobre el botón que represente a la unidad que desea crear, si tiene recursos suficientes el edificio procederá a construir la unidad especificada.

7.2.20. Cancelar Desarrollo

Diagrama de Secuencia:

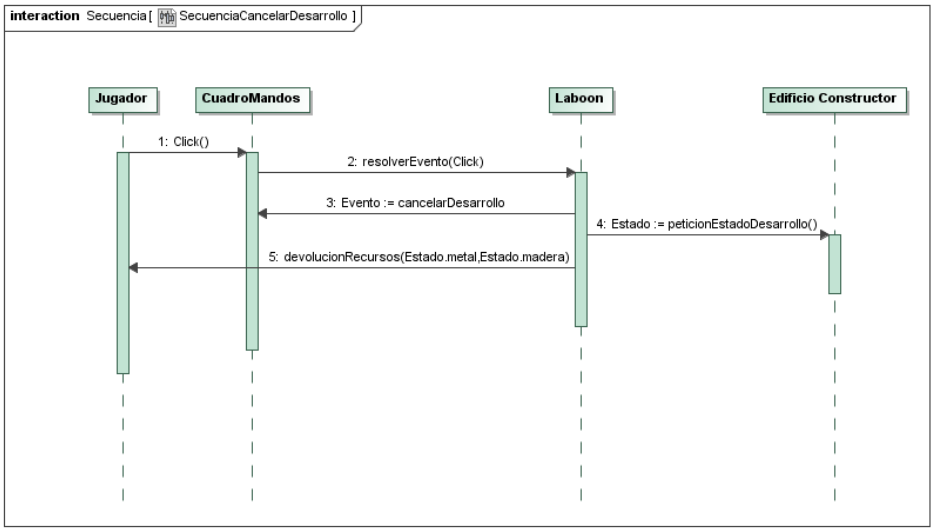

Figura 7.25: Diagrama Secuencia: Cancelar Desarrollo.

El jugador quiere cancelar el desarrollo de una tecnología o de creación de una unidad, para ello seleccionará el edificio correspondiente del cual desea cancelar el desarrollo y le dará al botón “Cancelar”, recuperando una parte proporcional al tiempo pasado de los recursos invertidos.

7.2.21. Comercio Recursos

Diagrama de Secuencia:

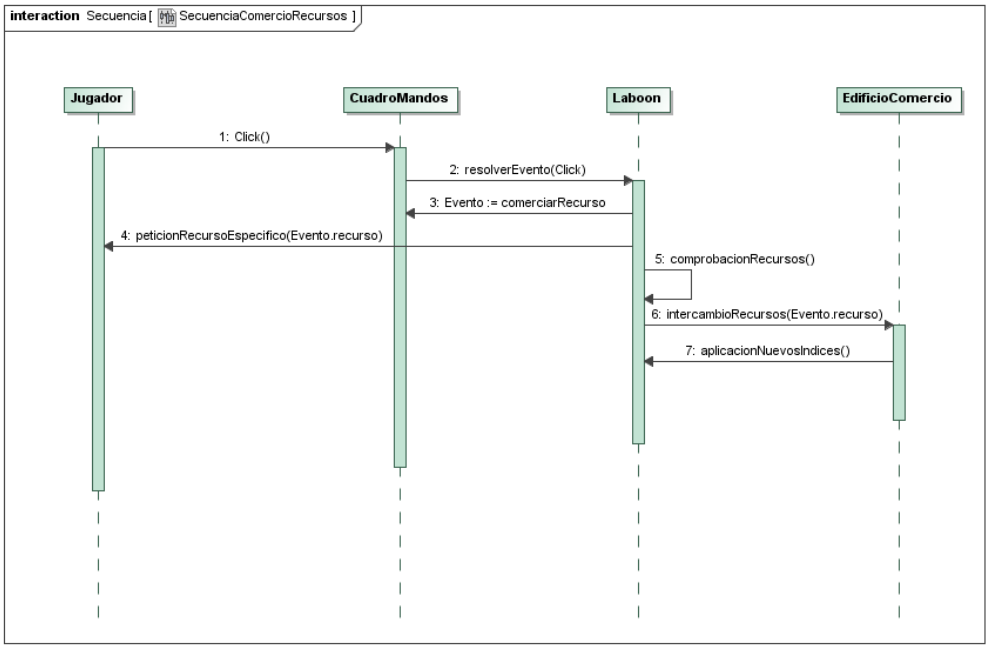

Figura 7.26: Diagrama Secuencia: Comerciar Recursos.

El jugador quiere intercambiar recursos, necesita recursos de un tipo y para ello debe entregar recursos del tipo contrario. Elegirá el botón del recurso que necesita y automáticamente se producirá el cambio de recursos, si tiene suficiente para realizar el intercambio.

7.2.22. Enviar Mensaje

Diagrama de Secuencia:

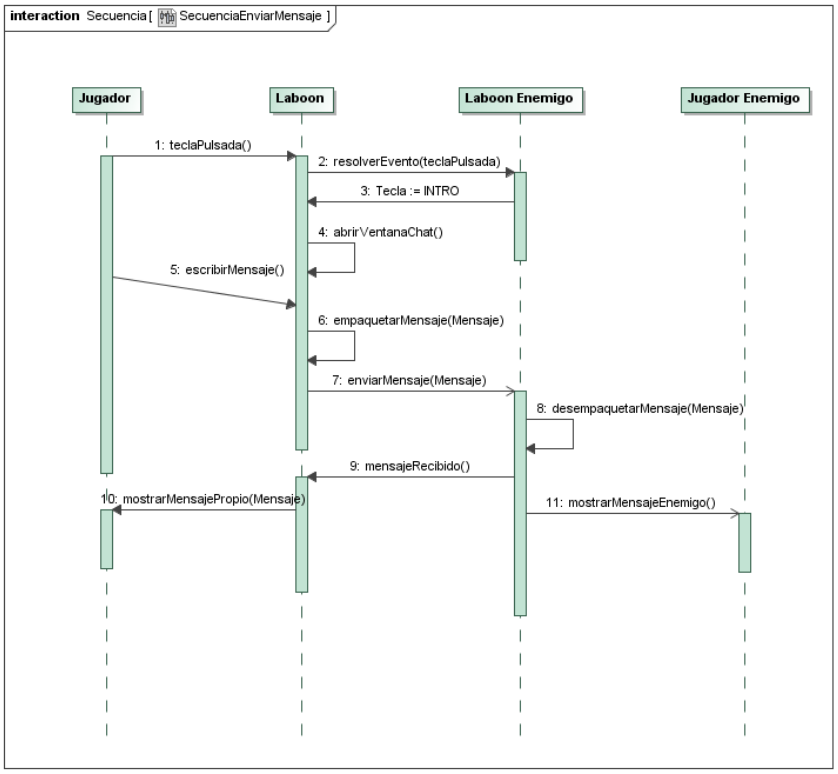

Figura 7.27: Diagrama Secuencia: Enviar Mensaje.

El jugador quiere enviar un mensaje al jugador contrario, para ello pulsará la tecla intro, escribirá el mensaje y volverá a pulsar la tecla intro para terminar de enviar el mensaje.

7.3. Clases de diseño

Para ver la documentación referente a las clases de diseño diríjase a la carpeta de documentación *doc* y abra el fichero **index.html**.

Capítulo 8

Implementación

En este capítulo se comentarán todos los aspectos técnicos relacionados con la implementación *Laboon*. Hablaremos desde las herramientas utilizadas para su desarrollo así como los problemas técnicos encontrados y las funcionalidades que proporciona esta aplicación.

8.1. Herramientas Utilizadas

Se han usado una serie de herramientas para desarrollar *Laboon*, tanto en Windows como en Linux, obviamente para cada sistema operativo se han usado herramientas distintas.

Software libre utilizado para la creación de la aplicación:

- Sistema Operativo Libre: Ubuntu 9.04 Jaunty Jackalope. Sistema operativo libre con licencia GNU/GPL.
- IDE de desarrollo: Eclipse usando C++.
- Bibliotecas LibSDL. Bibliotecas con licencia GNU/GPL.
- Modelado de figuras: Blender. Software con licencia GNU/GPL.
- Pintado 2D: Gimp. Software con licencia GNU/GPL.

Software adicional usado para la construcción de esta aplicación. El objetivo de usar estas aplicaciones es garantizar una compatibilidad de uso con los sistemas Windows, ya que son los más usados.

- Sistema operativo: Windows Vista Home Premium.
- IDE de desarrollo: Visual Studio Express 2008 usando C++.
- Pintado 2D: Aplicación de dibujado: Paint.Net.

Software utilizado para realizar la documentación:

- Editor de documentación: Latex.
- Compilador de documentación: TeXnicCenter.
- Editor de Diagramas: DrawMagic v16.6.
- Doxygen

8.2. Bibliotecas Gráficas Utilizadas

Elegimos libSDL frente a las otras dos. Las razones principales son que finalmente se decidió construir un juego bajo una plataforma 2D, lo cual hacía que las bibliotecas 3D perdieran gran parte de su capacidad y fueran descartadas. Otro objetivo que debía cumplir la biblioteca que fuéramos a escoger que fuera capaz de funcionar tanto en sistemas Windows como en sistemas Linux y a ser posible bajo licencia GPL, así que fueron descartadas otras bibliotecas tales con XNA.

En concreto se utilizaron: [10]

- SDL (Simple DirectMedia Layer). Conjunto de bibliotecas que proporcionan funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. [10]
- SDL_Image. Extiende notablemente las capacidades para trabajar con diferentes formatos de imagen.
- SDL_Net. Proporciona funciones y tipos de dato multiplataforma para programar aplicaciones que trabajen con redes.
- SDL_Mixer. Extiende las capacidades de SDL para la gestión y uso de sonido y música en aplicaciones y juegos.
- SDL_TTF. Permite usar fuentes *TrueType* en aplicaciones SDL.

8.3. Funcionalidades Implementadas

Laboon implementa una gran serie de funcionalidades para hacer que las partidas sean más sencillas para el jugador. Antes de comentarlas creo que es imprescindible conocer qué tipo de edificios y vehículos nos podremos encontrar durante una partida.

Edificios:

- Base Principal: Es el núcleo de operaciones de tu ejército. Incrementa la capacidad de población y la capacidad de almacenamiento de recursos
- Factoría Tanques: Construye tanques
- Factoría Artillería: Construye artilleros
- Puerto: Construye barcos
- Cámara de comercio: Intercambia recursos

- Laboratorio: Desarrollo e investigación de tecnologías
- Casa: Aumenta la capacidad de población
- Mina de metal: Recolecta metal
- Silo: Aumenta la capacidad de almacenamiento de recursos
- Torreta: Edificio defensivo que ataca a unidades enemigos

Vehículos militares:

- Artillero Ligero
- Artillero Pesado
- Tanque Ligero
- Lanzamisiles
- Destructor
- Incursor
- Emboscador
- Fragata

Vehiculos con funcionalidades especiales:

- Transporte: Carga en su interior 5 vehículos para transportarlos de forma segura
- Barco Recolector: Recolecta metal submarino
- Constructor: Construye y repara edificios
- Deforestador: Recolecta madera
- Excavador: Recolecta metal terrestre subterráneo

Una vez que hemos visto todos los vehículos y edificos que existen en *Laboon* podremos de una forma más sencilla explicar todas las funcionalidades que tiene implementadas. Comenzaremos por las sencillas.

- Recolección automática de recursos: Una vez que nosotros mandamos a un obrero a recolectar recursos, él automáticamente recolectará todos los recursos de la zona y los consuma por completo buscará una nueva zona de recolección
- Respuesta ante ataques enemigos: Si una de nuestras unidades es atacada por un vehículo enemigo, dicha unidad devolverá los ataques al agresor
- Respuesta de auxilio de unidades amigas: Si una de nuestras unidades es atacada emitirá un mensaje de auxilio hacia las unidades militares cercanas para socorrer y parar la agresión enemiga

- Capacidad de seleccionar varias unidades: *Laboon* permite seleccionar varias unidades y tratarlas todas como si fuera una sola
- Asignación de "rallypoint" a edificios: Existe la posibilidad de poner una marca de salida en una zona del tablero a los edificios que construyen unidades. Esta señal indica a dónde queremos que vayan las unidades una vez construidas
- Exploración terreno desconocido: Al principio el terreno del tablero es desconocido, debermos hacer uso de nuestras dotes de investigación para encontrar nuevas zonas con recursos, la base enemiga, zonas nuevas para expandirse, etc
- Construcción automática de unidades y edificios: Una vez que mandemos la orden de construir un vehículo o un edificio ya el jugador se puede desentender de la construcción hasta que haya finalizado
- Actualización automática de unidades y edificios tras investigaciones: Al realizar una nueva investigación de una tecnología, todos los vehículos y edificios sensibles a dichas mejoras, acutalizarán de forma automática sus parámetros y las nuevas unidades que se construyan tendrán ya sus parámetros actualizados

Dentro de *Laboon* existen muchas más funcionalidades que proporcionan muchas ayudas al usuario para poder controlar todo lo que está ocurriendo en el tablero de una forma rápida y más sencilla. La primera de estas funcionalidades es el uso del minimapa. Un minimapa es una representación a pequeña escala del tablero de juego el cual nos representa el estado actual del mismo. Con esta herramienta podremos ver todo lo que pasa en el tablero en tiempo real y poder actuar en consecuencia de una forma rápida y precisa.

Otra funcionalidad que posee *Laboon* es la capacidad de comunicarse con otro jugador a través de un sistema de chat.

Por otra parte también nos ofrece la posibilidad de crear nuestros escuadrones personalizados para poder manejar de forma rápida y sencilla varias unidades de una forma precisa.

Ahora vamos a hablar de dos edificios muy especiales que le dan mucha vida a *Laboon*. El primero de ellos es la cámara de comercio, este edificio nos da la oportunidad de comerciar recursos de forma instantánea con el sistema, pero cuidado, existe una bolsa de mercado, si la demanda de un tipo de recursos aumenta éste se volverá mas caro. Lo más interesante de este sistema de comercio, es que los índices de intercambio del mercado son compartidos por ambos jugadores, dando así la posibilidad de crear muchas estrategias.

El otro edificio del que hablábamos es el laboratorio. Este edificio, pone a nuestra disposición 4 tecnologías que pueden ser desarrolladas 3 veces incrementando su capacidad en cada nivel, son las siguientes:

- Ataque: Mejora el ataque de todas las unidades en un 10 % por nivel.
- Defensa: Mejora la defensa de todas las unidades en un 50 % por nivel.

Figura 8.1: Minimapa.

- Vida edificios: Mejora la vida de los edificios en un 15 % por nivel.
- Recolección: Mejora la velocidad de recolección de recursos en un 10 % por nivel.

Otra funcionalidad muy importante en *Laboon* es darle la capacidad a las unidades de esquivar obstáculos a la hora de moverse, dándoles así un comportamiento más inteligente, para ello se ha utilizado el algoritmo A* que es muy útil en problemas de búsqueda de caminos.

Hemos comentado anteriormente que *Laboon* es una aplicación exclusivamente multijugador aunque existe la posibilidad de poder jugar en modo entrenamiento contra el sistema. La arquitectura multijugador ha sido diseñada como una arquitectura P2P donde todos los usuarios son igualmente importantes. Se usa un testigo que pasa de un equipo a otro para decidir quién puede enviar información y quién tiene que recibir información. El equipo que tiene el testigo enviará toda la información referente a sus unidades y el equipo contrario se quedará a la escucha con una función de recepción bloqueante hasta que reciba toda la información así como el testigo. El sistema implementado es un sistema de microturnos que es prácticamente inapreciable para el jugador y le da sensación de estar jugando en tiempo real.

También se incorpora dentro de *Laboon* una banda sonora para que los jugadores puedan meterse más en la partida. De todas formas es posible que si a los jugadores no les gusta el tipo de canciones electrónicas puedan poner la música que ellos prefieran. Para ellos tan sólo deberán sustituir los archivos que se encuentran en la carpeta *music* por otros archivos de audio que contengan el mismo nombre.

A parte de música, también existen una serie de efectos sonoros en *Laboon* los cuales cumplen dos funciones. La primera de ellas es hacer que el juego sea más entretenido e interesante, ya que un juego sin efectos sonoros es un poco soso. Por otra parte el objetivo principal que cumplen son poder avisar al usuario de eventos que están ocurriendo en el tablero en zonas que él no esté mirando, consiguiendo que el usuario pueda responder de formar rápida y efectiva ante acciones realizadas por el enemigo.

Por último *Laboon* ofrece la capacidad de que el jugador pueda crearse sus propios mapas, con esto podemos conseguir que el juego tenga mucha más vida y que sea más interesante, tan solo deberán seguir la estructura contemplada en el manual de usuario que viene junto a este documento.

8.4. Problemas Encontrados

8.4.1. Imágenes

Durante el desarrollo de *Laboon* los problemas que han ido apareciendo no han sido pocos.

El principal problema que tenía al comenzar *Laboon* era el mismo problema que tiene toda persona al comenzar un nuevo proyecto, ¿por dónde comienzo?, bien aquí he de agradecer a mi director que me enviara documentación abundante y de muy buena calidad creada por Antonio García Alba al cual agradezco también su trabajo.

Una vez solventado cómo empezar a desarrollar el videojuego tenía un problema bastante grave, no tenía ningún recurso gráfico para utilizar, y un juego sin gráficos pierde mucho. Tras buscar bastante durante bastantes días por Internet dí con una página de un diseñador de videojuegos antiguos. Y tras ponerme en contacto con él, me comentó que podría usar sus imágenes sin problema, agradecimientos también a Daniel Cook. `http://lostgarden.com`.

Tras esto ya tenía recursos suficientes para desarrollar la aplicación, pero se presenta un nuevo problema, el formato de los ficheros de imagen obtenidos no es cómodo de usar con las bibliotecas SDL. Llegamos al punto en el que hay que estructurar todas las imágenes que se vayan a usar. Se tomó como decisión de diseño realizar una especie de *collage* a mano donde se encontrarían las diferentes casillas que compondrían el tablero de juego.

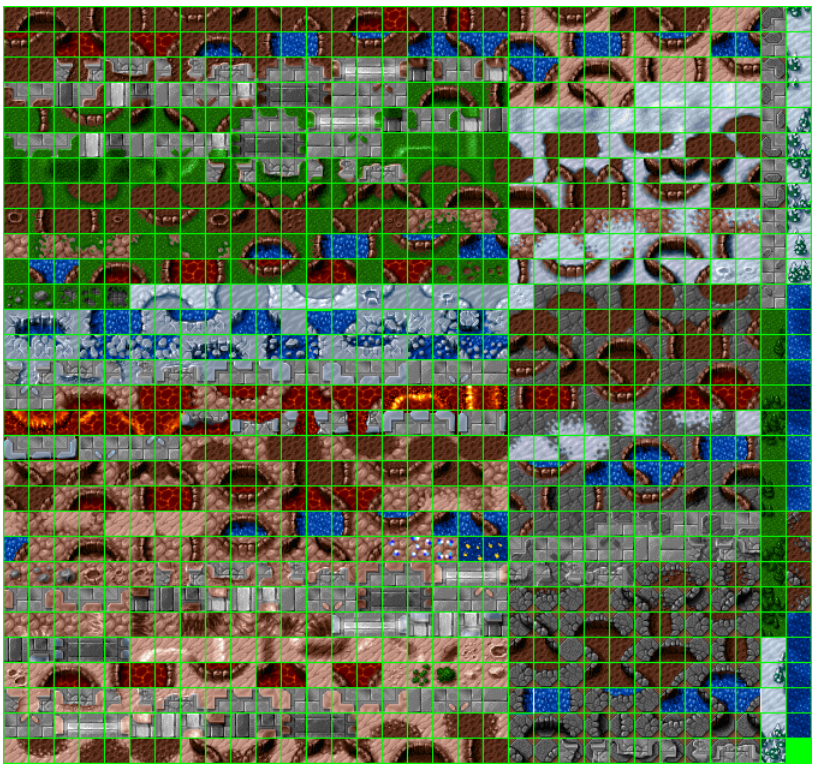

Figura 8.2: Casillas del tablero.

Una vez transformado todos los tipos de imágenes también tuve que realizar varias imágenes por cuenta propia para añadir cierta capacidad de asimilación al videojuego como son las tooltips. Una tooltip es un mensaje de ayuda que suele aparecer cuando pasamos el puntero del ratón por encima de un botón.

8.4.2. Pathfinding

Cambiando un poco el ámbito de los problemas nos centraremos en el algoritmo A*. Este algoritmo es realmente útil para los RTS que existen en el mercado, pero como todo hay que adaptarlo. Principalmente el algoritmo A* ha representado dos problemas bastante importantes.

El primero de ellos ha sido los tiempos de construcción de los árboles de búsqueda, al existir la capacidad de tratar con varios vehículos a la vez y al ser el tablero de juego tan extenso, a veces se producían varios árboles con más de cien niveles de profundidad produciendo que el juego se parase. El segundo problema existente era que el cálculo de los caminos a recorrer solo se realizaba a la hora de dar la órden correspondiente (mover, atacar, construir, reparar), por tanto si en mitad del recorrido se interponía un obstáculo móvil, tal como un vehículo que se encuentra cruzando, uno de los dos vehículos desaparece ya que machaca al otro.

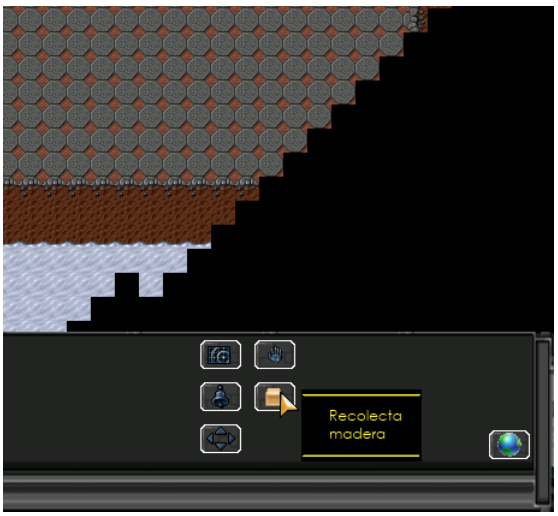

Figura 8.3: Ejemplo de tooltip.

Para arreglar estos problemas primeramente tuve que implementar el algoritmo A* modificado para calcular los caminos en varias iteraciones consiguiendo un mejor tiempo de respuesta ante el usuario, que es la prioridad número uno. Por otra parte, se realizaron modificaciones sobre dicho algoritmo para poder localizar obstáculos dinámicos en el camino y poder realizar acciones correctivas al respecto, como pararse o buscar un camino alternativo.

8.4.3. Multijugador

En el aspecto multijugador también se tuvieron que realizar varios cambios debidos a varios problemas encontrados durante las pruebas a través de LAN y de Internet. El primer problema encontrado fue que el consumo de ancho de banda era bastante importante, estamos hablando de unos 120KBps de subida y de bajada, algo descomunal para las líneas actuales en España por lo menos. Para solucionar esto, básicamente se tuvo que reestructurar el sistema multijugador y crear un sistema de codificación de mensajes para reducir el consumo de ancho de banda a costa de aumentar un poco la carga sobre la CPU, pero el cambio fue bastante satisfactorio. Se redujo el consumo a unos 10-12KBps de subida y bajada y los tiempos de respuesta aumentaron apenas unos 20 milisegundos.

El otro gran problema encontrado en el sistema multijugador fue el hecho de que todas las acciones de *Laboon* se realizaban basadas en frames. Un frame es una unidad de tiempo que se basa en el la velocidad en la que la CPU consigue dar una vuelta completa al bucle principal del juego. En principio para una partida de un jugador está muy bien y es perfectamente válido, pero en partidas multijugador puede provocar problemas de sincronización, ya que si dos equipos tienen diferentes velocidades de reloj, esto produce que una acción realizada tanto en un equipo como en otro vaya a diferentes “velocidades” produciendo errores de sincronización graves en el tablero. No está permitido que una unidad esté en diferentes posiciones en cada uno de los tableros.

Este problema se corrigió dándole a todas las acciones que fueran sensibles a estos posibles fallos de sincronización unas unidades de tiempo universales, tales como los milisegundos. Aunque un equipo sea más lento que otro equipo éste será capaz de poder llevar a la vez todas las unidades de ambos jugadores sin producir errores de sincronización.

8.5. Estadísticas

Tras la realización de la apliación *Laboon* se realizaron una serie de estadísticas sobre el proyecto que pueden resultar interesantes.

Se han usado distintos programas (tortoiseSVN y CCCC) para obtener datos sobre las estadísticas de este proyecto.

- LDC: 47.026
- LDC media por módulo: 505,656
- LDComentarios: 11.268
- LDComentarios media por módulo: 121,161
- Nº de módulos: 93

Aparte de las estadísticas del propio proyecto también se han obtenido diferentes estadísticas del sistema de control de versiones.

- Commits realizados : 295
- Ficheros modificados: 5799

Para terminar adjuntamos una gráfica con la actividad (basada en commits) del proyecto:

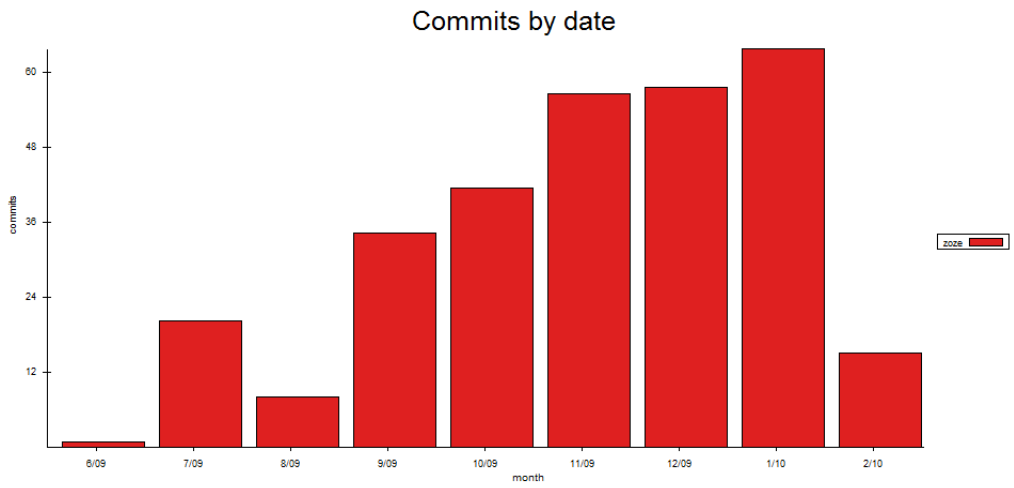

Figura 8.4: Estadísticas Commits Laboon.

Capítulo 9

Pruebas y Validaciones

El desarrollo de un plan de pruebas es fundamental para completar la finalización de cualquier aplicación. Laboon interacciona tanto con el sistema operativo del jugador como con el del contrincante, a parte de interaccionar con sus propias interfaces a través de Internet. El plan de prueba consiste en realizar un gran número de pruebas llevando al programa a los puntos límites donde podría fallar.

Las pruebas que realizaremos sobre la aplicación son:

- Pruebas de contenido: Estas pruebas nos permiten detectar errores en el contenido de la aplicación.
- Pruebas de interfaz: Estas pruebas validan los aspectos relativos al funcionamiento de la interfaz y como responde ante los jugadores, así como la forma de mostrarles la información de sus unidades y las acciones que han realizado.
- Pruebas de navegación: Estas pruebas nos sirven para comprobar, que podemos acceder a todos los menús y podemos realizar todos los cambios de configuración que queramos sin ningún tipo de error.
- Pruebas de componentes: Estas pruebas comprueban de una forma algo más general que tanto Laboon en modo un jugador como en modo multijugador, cumplen todas sus especificaciones dentro de los caso de usos descritos con anterioridad.

9.1. Plan de Pruebas

9.1.1. Introducción

Durante el plan de pruebas, *Laboon* será sometido a diferentes pruebas, divido en menús y juego.

9.1.2. Test a realizar

Se someterán a test los siguientes elementos.

- Menu Laboon.
- Código Fuente menu Laboon.
- Juego Laboon.
- Código Fuente juego Laboon.
- Multijugador Laboon.
- Código Fuente Multijugador Laboon.

9.1.3. Aspectos a ser probados

Ahora listaremos que aspectos quedarán sometidos bajo los test a realizar.

Por parte de menu Laboon.

- Pruebas de Configuración:
  - Test de cambio de Resolución.
  - Test de cambio de aceleración.
  - Test de activación/desactivación de audio.
  - Test de activación/desactivación de tooltips.
- Test inicio Partida Un Jugador:
  - Elección Mapa.
  - Elección Raza.
- Test inicio Partida Multijugador:
  - Introducir Nombre.
  - Elección Mapa.
  - Elección Raza.
  - Unión a partida creada.
  - Lanzamiento partida multijugador.
- Test Juego:
  - Mover Unidades.
  - Atacar Unidades.
  - Construir Unidades.
  - Recolectar Recursos.
  - Construir Edificios.
  - Reparar Edificios.

9.1.4. Pruebas realizadas

- Validación código Menú Laboon.

  Descripción: Se comprueba que el código referente al menú de *Laboon* es correcto y no contiene fallos de ningún tipo.

  Resultados: Corrección de errores, borrado de variables no usadas, reestructuración de funciones, optimización de código.

- Validación código Juego Laboon.

  Descripción: Se comprueba que el código referente al juego de *Laboon* es correcto y no contiene fallos de ningún tipo.

  Resultados: Corrección de errores, borrado de variables no usadas, reestructuración de funciones, optimización de código.

- Test de interfaz con el sistema operativo.

  Descripción: Se comprueba con el sistema operativo y la aplicación por el uso de las acciones de petición de imágenes e información (por ejemplo construcción del minimapa) son correctas, devolviendo la información esperada.

  Resultados: Control más exhaustivo de excepciones e información más detallada en ciertos puntos.

- Test de interfaz de usuario.

  Descripción: Se comprueba si el interfaz proporciona todas las funciones que se describen dentro de los casos de uso.

  Resultados: Se obtuvo un resultado satisfactorio.

- Test de Portabilidad.

  Descripción: Se comprueba si el programa es soportado por varias máquinas con distintos sistemas operativos.

  Resultados: Se acotan e incrementan los requisitos iniciales necesarios para lanzar la aplicación.

- Test de aplicación de órdenes.

  Descripción: Coprobación de que todas las funciones del jugador se ejecutan de forma correcta y que en una partida multijugador se mandan al sistema contrario.

  Resoltados: Se obtuvieron resultados satisfactorios.

Parte III

Manuales, conclusiones y apéndices

Capítulo 10

Manuales

10.1. Manual de Instalación y Configuración

10.1.1. Windows

Para jugar a *Laboon* sólo tendremos que descargarnos la versión correspondiente a Windows y descomprimirla. Tras descomprimir el archivo en una carpeta destino, solo deberemos ejecutar el archivo **Laboon.exe** para comenzar a jugar.

10.1.2. Linux: Ubuntu 9.10

Para instalar *Laboon* en Linux, deberemos descargarnos la versión correspondiente a Linux y descomprimirla. Una vez descomprimida deberemos bajarnos las bibliotecas necesarias. Para ello abriremos el gestor de paquetes *Synaptic*.

Figura 10.1: Instalación Linux: Paso 1.

Una vez abierto *Synaptic* pincharemos en **Buscar** e introduciremos *SDL* en el cuadro de búsqueda.

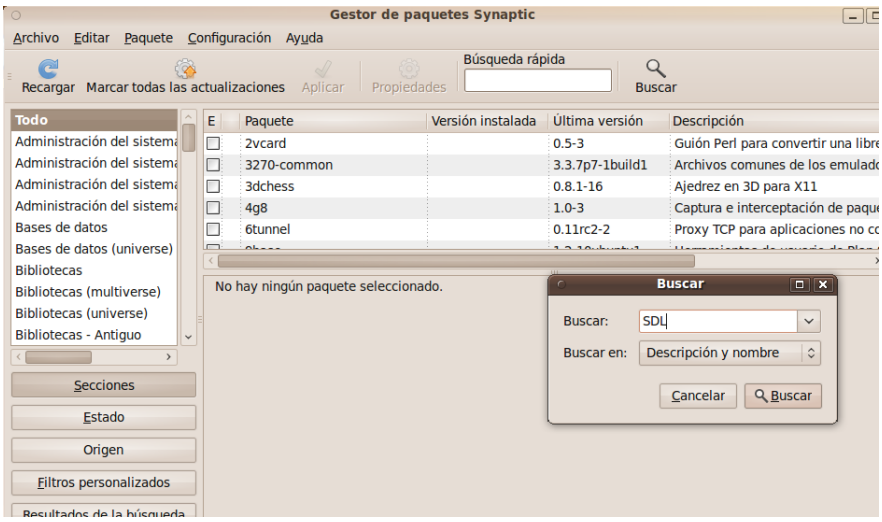

Figura 10.2: Instalación Linux: Paso 2.

Tras esto, esperaremos unos segundos y aparecerá una lista de paquetes relacionados con *SDL*. Seleccionaremos los siguientes:

- *libsdl1.2-dev*
- *libsdl-image1.2-dev*
- *libsdl-mixer1.2-dev*
- *libsdl-net1.2-dev*
- *libsdl-ttf2.0-dev*

Una vez seleccionados esos paquetes, de forma automátiuca se seleccionarán varios paquetes adicionales dependientes de los seleccionados.

newpage

Figura 10.3: Instalación Linux: Paso 3.

Una vez instalado todo, tan sólo ejecutaremos el archivo *laboon* en la carpeta donde hayamos descargado nuestra versión del mismo,

10.1.3. Una vez instalado...

Una vez lanzada la aplicación podremos acceder a **Opciones** para configurar los distintos parámetros del videojuego.

- Resolución: El usuario podrá elegir entre 3 posibles resoluciones: 800x600, 1024x768, 1280x1024. El usuario elegirá la que se ajuste a sus necesidades y luego pulsará el botón "Guardar".
- Aceleración: El usuario podrá activar o desactivar la aceleración Hardware. El usuario decidirá si quiere activarla o no y luego pulsará el botón "Guardar".
- Sonidos: El usuario podrá activar o desactivar los sonidos del videojuego. El usuario decidirá si quiere activarlos o no y luego pulsará el botón "Guardar".
- Subtítulos: El usuario podrá activar o desactivar los subtítulos que aparezcan en el videojuego. El usuario decidirá si quiere activarlos o no y luego pulsará el botón "Guardar".

10.2. Manual de Usuario

El manual de usuario se encuentra en el fichero **ManualDeUsuario.pdf**. Abra dicho fichero para ver cómo jugar a *Laboon*.

10.3. Manual del Desarrollador

Como se ha comentado anteriormente *Laboon* es un juego libre bajo GPL y por tanto cualquier persona puede continuar el trabajo por su cuenta. En este punto mostraremos cómo configurar nuestro entorno de desarrollo para seguir con *Laboon* tanto en Windows como en Linux, distribución Ubuntu.

10.3.1. Windows

Lo primero que deremos hacer es descargarnos las bibliotecas necesarias de desarrollo para SDL. Para ello lo primero que haremos es abrir nuestro navegador e irnos a la página de `http://www.libsdl.org`. Una vez abierta esta página pincharemos en *SDL 1.2* dentro de la sección Downloads.

Una vez aquí elegiremos la versión para VC-8 para Visual C++ 2005, tal y como indica la figura.

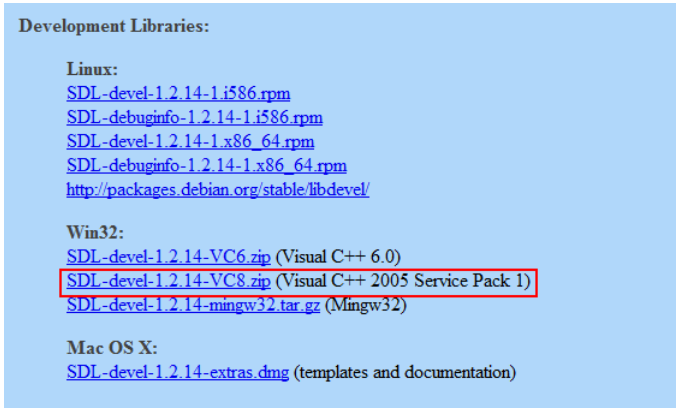

Figura 10.4: Descarga Bibliotecas SDL Windows

Una vez descargada estas bibliotecas, nos descargaremos las bibliotecas adicionales para tener la funcionalidad completa de SDL.

- SDL_image: `http://www.libsdl.org/projects/SDL_image/`
- SDL_mixer: `http://www.libsdl.org/projects/SDL_mixer/`
- SDL_net: `http://www.libsdl.org/projects/SDL_net/`
- SDL_ttf: `http://www.libsdl.org/projects/SDL_ttf`

De cada una de estas direcciones nos vamos a bajar las partes devel VC6 de la sección Binary.

Una vez tenemos todas las bibliotecas necesarias nos crearemos una carpeta dentro de nuestro disco duro QUE NO BORRAREMOS, así que cuidado con donde los ponemos. En mi caso me he creado una carpeta llamada *Adicionales* en el disco duro C. y he descomprimido todos los ZIP de las bibliotecas adicionales, y luego me he creado una carpeta llamada *SDL-1.2.13* donde tengo la biblioteca principal de SDL.

Figura 10.5: Descompresión bibliotecas SDL.

Ahora nos dispondremos a abrir nuestro Visual Studio para proceder a su configuración. Esta configuración consta de 2 partes:

- Configuración del entorno: Inclusión de los ficheros .h y los .lib.
- Configuración del proyecto: Inclusión de los archivos .dll a usar.

Una vez abierto nuestro editor nos iremos a la barra de menú superior y le daremos a Herramientas - Opciones. Una vez abierto seleccionaremos *Proyectos y Soluciones* y dentro *Directorios de VC++*.

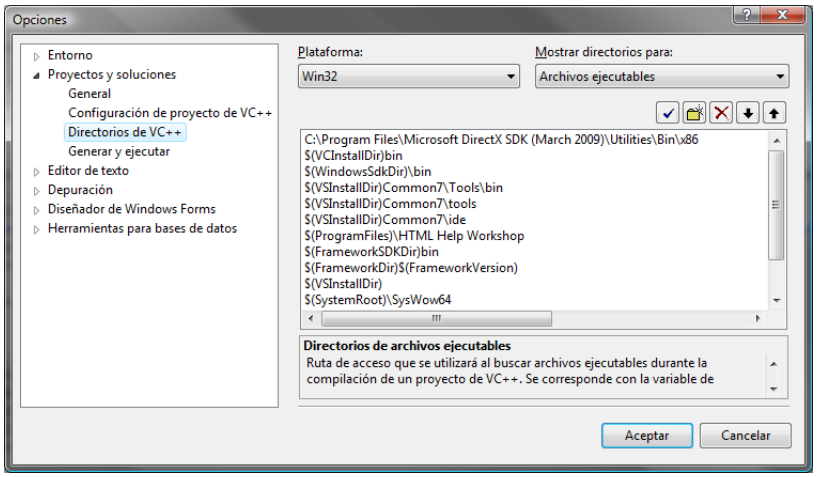

Figura 10.6: Configuración Visual Studio. Paso 1.

Una vez aquí nos iremos al menú desplegable de la derecha y elegiremos la opción de **archivos de inclusión** y puslaremos sobre el icono de una carpeta amarillo. Al darle a este icono aparecerá un nuevo elemento en la lista y tras pulsar en ... podremos elegir las carpetas “Include” de SDL. Estas carpetas son todas las carpetas include de cada uno de los archivos descomprimidos. Una vez todos incluidos debería quedar algo así.

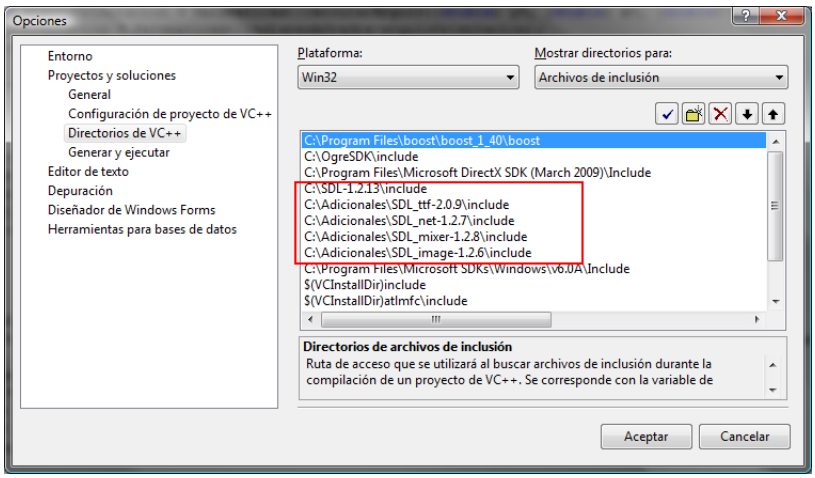

Figura 10.7: Configuración Visual Studio. Paso 2. Inclusión de archivos de cabecera.

Una vez hecho esto, repetiremos el proceso con los archivos .lib. Escogeremos la opción de **archivos de biblioteca** e incluiremos todas las carpetas LIB de las bibliotecas SDL. Debería quedar algo así.

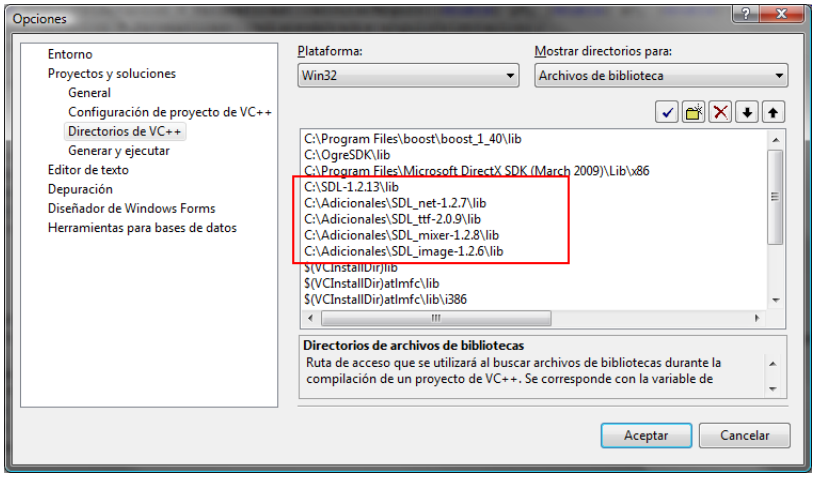

Figura 10.8: Configuración Visual Studio. Paso 3. Inclusión de archivos de biblioteca.

Bien, el primer paso ya está completo, ahora deberemos abrir nuestro proyecto de Visual Studio que contiene a *Laboon* mediante su correspondiente archivo .sln. Tras esto le daremos a botón derecho sobre el nombre del proyecto y nos iremos a propiedades.

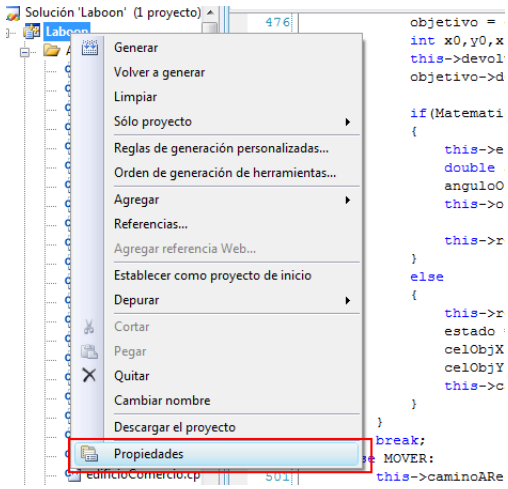

Figura 10.9: Configuración Visual Studio. Paso 4. Inclusión de LIBs al proyecto (1 de 2).

Una vez hecho esto nos saldrá una nueva ventana donde incluiremos los archivos dll a nuestro proyecto. Para eso nos iremos a la opción *Vinculador* y dentro de éste la opcion *Entrada*. En la línea **Dependencias adicionales** introduciremos los siguientes archivos separados por espacios. Y en la parte superior elegiremos la opción en *Configuración* llamada Release.

- sdl.lib
- sdlmain.lib
- sdl_image.lib
- sdl_net.lib
- sdl_ttf.lib
- sdl_mixer.lib

El resultado final debería ser algo así:

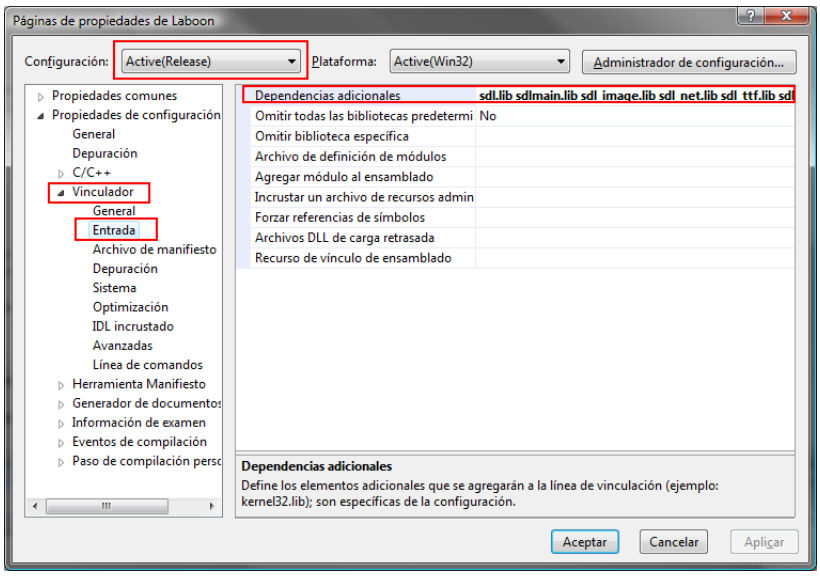

Figura 10.10: Configuración Visual Studio. Paso 5. Inclusión de LIBs al proyecto (2 de 2).

10.3.2. Ubuntu

Primero de todo se recomienda que se sigan los pasos en el apartado 10.2 para instalar las bibliotecas necesarias para poder desarrollar *Laboon.*

Una vez descargadas podremos empezar a desarrollar *Laboon.* Para poder desarrollar *Laboon* bajo linux deberemos tener instalado el editor Netbeans para C++. Para ello deberemos irnos a su página oficial `http://www.netbeans.org`. Una vez dentro nos iremos al botón Download Netbeans 6.8 Free y nos saldrán todos los netbeans disponibles. Elegiremos Netbeans para C++.

Figura 10.11: Descarga Netbeans.

Una vez descargado Netbeans, tan solo tendremos que ejecutarlo como si fuera un archivo .sh e indicarle donde tenemos instalada nuestra JVM para

poder ejecutarla.

Una vez instalado comenzaremos a configurar Netbeans para el desarrollo de SDL. Para ello abriremos nuestro proyecto en netbeans y nos aseguraremos de que todas las bibliotecas están incluidas en el proyecto. El primer paso a realizar es irnos a las propiedades de nuestro proyecto, tal como indica la figura.

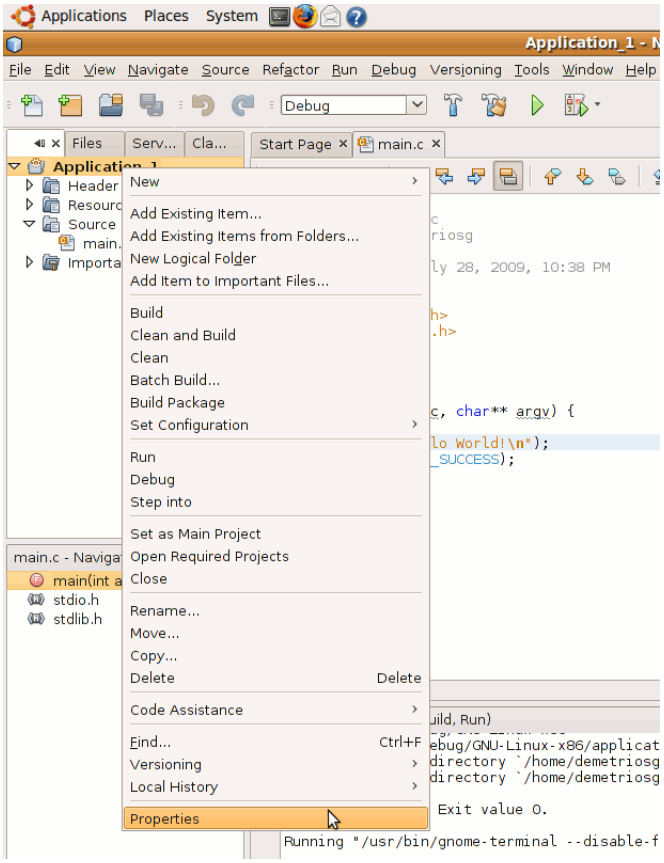

Figura 10.12: Configuración Netbeans. Paso 1.

Tras esto se nos abrirá una venta, sobre la cual pincharemos en la opción **C++ Compiler** y en Configuración elegiremos **All Configurations**.

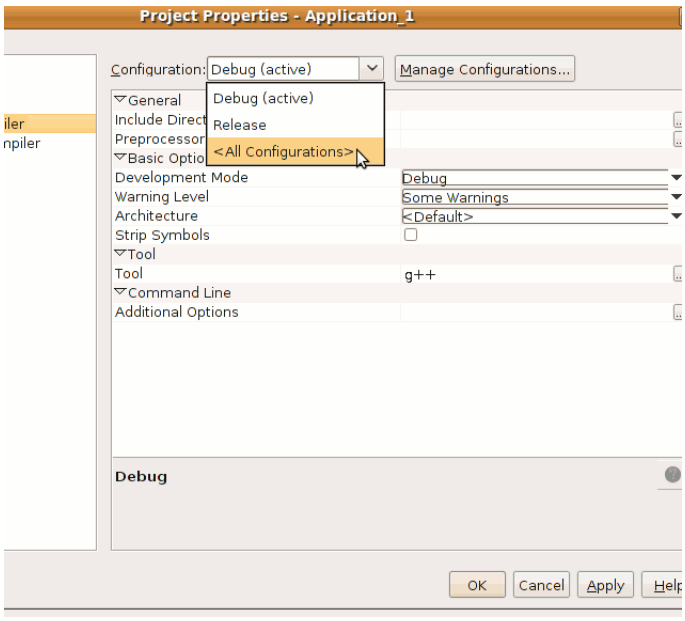

Figura 10.13: Configuración Netbeans. Paso 2.

Una vez hecho esto le daremos sobre el botón con etiqueta "..." sobre **Include Directories** y luego pulsaremos el botón **Add**.

Se nos abrirá una nueva ventana donde tendremos que decir donde está instalado SDL. Nos iremos navegando sobre las distintas carpetas hasta llegar a /usr/include/SDL y pulsaremos la opcion "Absolute".

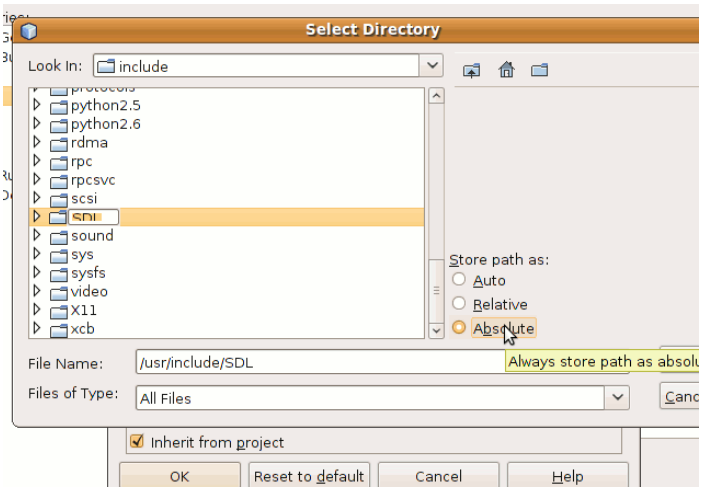

Figura 10.14: Configuración Netbeans. Paso 3.

Tras esto le daremos a aceptar y volveremos a la ventana anterior, ahora nos iremos a la opción **Linker** y volveremos a cambiar la Configuración a **All Configurations**. Ahora pulsaremos en el botón con etiqueta "..." sobre **Libraries** y en la nueva ventan le daremos a **Add Library**.

Ahora nos volverá a salir una lista de carpetas y navegaremos hasta /usr/lib/ donde añadiremos el fichero *libSDL.a*, tal como indica la siguiente figura.

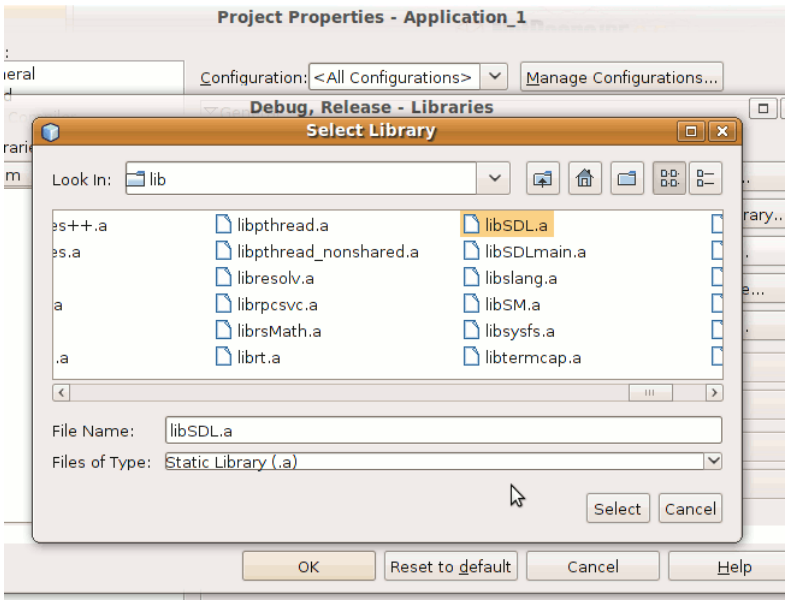

Figura 10.15: Configuración Netbeans. Paso 4.

Repetiremos este proceso añadiendo los siguientes ficheros:

- libSDLmain.a
- libSDL_image.a
- libSDL_net.a
- libSDL_ttf.a
- libSDL_mixer.a

Guardamos y aceptamos todo y terminamos la configuración.

10.3.3. Diferencias código Linux y Windows

La única diferencia entre los códigos de *Laboon* de Windows y Linux, es la funcion *Abs()* en sistemas Linux esta función no recibe parámetros de tipo double mientras que en Windows sí

Capítulo 11

Conclusiones

La primera conclusión que podemos sacar es que *Laboon* ha cumplido todos los objetivos propuestos inicialmente.

1. Crear un videojuego de tipo RTS 100 % funcional
2. Crear un videojuego multiplataforma
3. Capacidad de personalización de los aspectos técnicos del videojuego
4. Capacidad de jugar solo para entrenar y aprender
5. Capacidad de jugar contra otras personas a través de los protocolos LAN o TCP/IP
6. Capacidad de comunicación a través de un sistema de chat

Cumple todas las bases de un juego tipo *RTS*. Proporciona un modo de juego **Un Jugador** para que el usuario practique el sistema del juego y también proporciona la capacidad de jugar partidas multijugador usando el protocolo TCP/IP a través de la red.

Otra funcionalidad adicional que posee *Laboon* es la capacidad de que el usuario pueda crearse sus propios mapas aplicando las restricciones indicadas en la sección 10.2.

Además el usuario podrá personalizar la configuración de su propia copia de *Laboon* así como crear su propia lista de reproducción que sonará mientras esté jugando también indicado en la sección 10.2.

Durante todo el proceso de desarrollo de *Laboon* las dificultades encontradas no han sido pocas, por una parte tenemos el desconocimiento de las bibliotecas usadas para el desarrollo de videojuegos, que me llevaron varias semanas dominarlas, pero el problema no se quedó ahí. Hay muchos aspectos en el mundo de los videojuegos a tener en cuenta, dependiendo del tipo de juego que se vaya a realizar el enfoque es totalmente distinto, no tiene nada que ver un RTS con un FPS o con un juego de plataformas.

Otra gran dificultad encontrada ha sido que la dimension del proyecto ha sobrepasado con creces las espectativas iniciales debido al desconocimiento del

campo que estamos tratando y por otro lado nos encontramos que necesitamos una serie de recursos de las cuales a priori no disponemos de ellas, como los gráficos y los sonidos (imprescindibles para cualquier videojuego).

Uno de los mayores problemas que se pueden crear sobre como enfocar el sistema del juego, han sido aplacados fácilmente debido al extenso conocimiento poseido en juegos de varias categorias nombradas anteriormente en este documento.

Aunque *Laboon* es un videojuego completo hay muchos aspectos mejorables:

- Todo el entorno gráfico se podría mejorar notablemente con la ayuda de algún diseñador o grafista que se encargara de todos los aspectos visuales del videojuego.
- Respecto a los bandos del videojuego se aumentarán las diferencias para personalizarlas más. Otro aspecto a mejorar será aumentar la cantidad de jugadores que podrán jugar en una partida, tanto a nivel de **Un Jugador** como en **Multijugador**.
- Otro aspecto dentro del videojuego sería mejorar el *Lore* del mismo, es decir crear una historia con un trasfondo para añadirle al videojuego un poco de consistencia. Así como la posibilidad de añadirle inteligencia artificial para poder jugar partidas largas como campañas.
- Por último se podría crear una aplicación paralela al videojuego, un *Map Editor*, para facilitar al usuario crear los mapas del juego y poder imponer sus propias reglas al mismo. Con esto se conseguiría que el usuario creara sus propias fases y campañas.

Aunque las dificultadas encontradas y las horas aplicadas a este proyecto han sido muchas, tras la finalización de este proyecto he comprendido que hubiesen sido necesarias varias personas para realizar el videojuego debido a la complejidad que ello conlleva, aunque ello conlleve mantener canales de comunicacion activos que "aumenten" los costes finales para evitar problemas de desinformación. Por otra parte aunque mucha gente quiera dejarlo de lado la potencia que proporciona C++ es enorme y muchas veces el problema está más en el programador que en la herramienta. Otra cosa cosa que aprendido durante la realización de este proyecto ha sido que a veces no hay que buscar el equilibrio entre tiempo y espacio, ya que por ejemplo en este tipo de juegos el tiempo de respuesta tiene muchísima más preferencia que ahorrar espacio en memoria, algo a lo que no estaba habituado hasta ahora.

Para terminar, sólo comentar que al ser este juego software libre pueda ayudar a otras personas a desarrollar aplicaciones y así facilitar su aprendizaje, personalmente no he encontrado casi ninguna referencia de como programar muchos aspectos necesarios y gracias a comentarios en distintas comunidades como **GameDev** y mucho esfuerzo, he podido resolverlas. Con esta apliacion esperamos aportar nuestro granito de arena a la comunidad para allanar el camino a otras personas.

Bibliografía

[1] Antonio García Alba, *Tutorial para la programación de videojuegos*, `http://softwarelibre.uca.es/tutorialSDL.html` Enero 2008. Recurso Electrónico.

[2] Brian W.Filzpatrick, C.Michael Pilato *Version Control With Subversion*, ISBN: 0-596-0048-6 `http://books.google.es/books?id=tMpR77ZeHhMC&printsec=frontcover&dq=svn+book+isbn&source=bl&ots=NsSGfbo2Cq&sig=R-fZzfyUGFvY38aMt51ryeomYF0&hl=es&ei=amtlS6ChE8q6jAeXsZWWBw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CAkQ6AEwAA#v=onepage&q=&f=false`

[3] Steve Rabin, *AI Game Programming Wisdom*, ISBN: 1-58450-077-8 `http://books.google.es/books?id=4f5Gszjyb8EC&pg=PA3&lpg=PA3&dq=strategy+video+game+programming+ISBN&source=bl&ots=9AVMuqSCqc&sig=LKod7cMNx5fMOdnca59fNULQ0z4&hl=es&ei=4EtlS9LOFoTw0wTCtJjJBg&sa=X&oi=book_result&ct=result&resnum=8&ved=0CC4Q6AEwBw#v=onepage&q=&f=false`

[4] Mark DeLoura *Game Programming Gems 2*, ISBN: 1-58450-054-9 `http://books.google.es/books?id=M2QYbTVd0VgC&pg=PA257&lpg=PA257&dq=strategy+video+game+programming+ISBN&source=bl&ots=K4sXgywMZP&sig=kAFq1IsMSGoUHUKxaYf6TvXtNYg&hl=es&ei=4EtlS9LOFoTw0wTCtJjJBg&sa=X&oi=book_result&ct=result&resnum=10&ved=0CDUQ6AEwCQ#v=onepage&q=&f=false`

[5] Glenford J. Myers *The Arto of Software Testing*, ISBN: 0-471-04328-1 `http://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/0471043281`

[6] Gerardo Aburruzaga García, Inmaculada Medina Bulo, Francisco Palomo Lozano, *Fundamentos de C++*, ISBN: 84-7786-734-8 Servicio de Publicaciones de la Universidad de Cádiz 2001.

[7] H.M. Deite, P.J. Deitel, *Cómo Programar en C/C++ 2ª Ed*, ISBN: 0-13-226119-7, Prentince Hall 1995.

[8] Stuart Rusell, Peter Norvig, *Inteligencia Artificial: Un enfoque moderno*, Prentice Hall 2004.

[9] S. Pressman, Roger *Ingeniería del Software: Un enfoque práctico* ISBN: 84-4813-214-9 Mac Graw Hill, 2003

[10] http://es.wikipedia.org

[11] http://indicelatino.com/juegos/historia/

[12] http://www.tecnologiahechapalabra.com/sociedad/sociologia_comunicacion/articulo.asp?i=1718

[13] http://www.palermo.edu/economicas/pdf_economicas/pdfs_centrodeentretenimientosymedios/Articulo_videojuegos_v4.pdf

[14] http://cgnauta.blogspot.com/2009/01/microsoft-alcanza-las-28-millones-de.html

[15] http://www.localarea.org/wiki/mccnm336/rts?wikiPageId=202872

[16] http://www.libsdl.org

[17] http://www.wired-weasel.com/users/serhid/blog/?page_id=4

[18] http://www.viti.es/gnu/licenses/gpl.html

[19] http://www.doxygen.org

Apéndice A

Licencia GPL v3

A.1. Licencia GPL

A.1.1. GNU GPL

La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Publical License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre de GNU (GFDL), la Open Audio License, para trabajos musicales, etcétera, y otras menos restrictivas, como la MGPL, o la LGPL (Lesser General Publical License, antes Library General Publical License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

A.1.2. Validez Legal

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se la denomina contrato de licencia o acuerdo de licencia.1 2 En los países de tradición anglosajona existe una distinción doctrinal entre licencias y contratos, pero esto no ocurre en los países de tradición civil o continental. Como contrato, la GPL debe cumplir los requisitos legales de formación contractual en cada jurisdicción. La licencia ha sido reconocida por juzgados en Alemania, particularmente en el caso de una sentencia en un tribunal de Munich,3 lo que indica positivamente su validez en jurisdicciones de derecho civil.

A.2. GPL

Aquí presentamos la traducción al español de la licencia GPL "GNU Public License", versión 3 que cubre la mayor parte dl software de la Free Software Foundation, y muchos más programas.

IMPORTANTE: Esta traducción tiene meramente caracter informativo y carece de validez legal. Si distribuye software libre o ha recibido software libre con esta traducción utilice para cuestiones legales la GPL versión 3 en inglés de la Free Software Foundation. [18]

A.2.1. GNU GENERAL PUBLIC LICENSE(GPL)

Versión 3, 29 de junio de 2007

Copyright (C) 2007 Free Software Foundation, Inc. ¡http://fsf.org/¿

Se permite la copia y distribución de copias literales de esta licencia, pero no está permitido modificarla.

A.2.2. *Preámbulo*

La Licencia Pública General GNU (GNU GPL) es una licencia libre, sin derechos para software y otro tipo de trabajos.

Las licencias para la mayoría del software y otros trabajos prácticos están destinadas a suprimir la libertad de compartir y modificar esos trabajos. Por el contrario, la Licencia Pública General GNU persigue garantizar su libertad para compartir y modificar todas las versiones de un programa–y asegurar que permanecerá como software libre para todos sus usuarios. Nosotros, La Fundación de Software Libre, usamos la Licencia Pública General GNU para la mayoría de nuestro software; y también se aplica a cualquier trabajo realizado de la misma forma por sus autores. Usted también puede aplicarla a sus programas.

Cuando hablamos de software libre, nos referimos a libertad, no a precio. Nuestras Licencias Públicas Generales están destinadas a garantizar la libertad de distribuir copias de software libre (y cobrar por ello si quiere), a recibir el código fuente o poder conseguirlo si así lo desea, a modificar el software o usar parte del mismo en nuevos programas libres, y a saber que puede hacer estas cosas.

Para proteger sus derechos, necesitamos evitar que otros le nieguen esos derechos o le pidan renunciar a ellos. Por lo tanto, usted tiene ciertas responsabilidades cuando distribuye copias del software, o si lo modifica: responsabilidades que persiguen respetar la libertad de otros.

Por ejemplo, si distribuye copias de tales programas, gratuitamente o no, debe transmitir a los destinatarios los mismos derechos que usted recibió. Debe asegurarse que ellos también reciban o puedan conseguir el código fuente. Y debe mostrarles estos términos y condiciones para que conozcan sus derechos.

Los desarrolladores que usen la GPL GNU protegen sus derechos de dos formas: (1) imponen derechos al software, y (2) le ofrecen esta Licencia para que legalmente lo copie, distribuya y/o modifique.

Para proteger a desarrolladores y autores, la GPL expone claramente que no existe garantía alguna para este software libre. Para beneficio de ambos, usuarios y autores, la GPL establece que las versiones modificadas deberán estar identificadas como tales, para que cualquier problema no sea atribuido por error a los autores de versiones anteriores.

Algunos dispositivos están diseñados para negar al usuario la instalación o la ejecución de versiones modificadas del software que usan internamente, aunque el fabricante sí pueda hacerlo. Esto es completamente incompatible con el objetivo de proteger la libertad de los usuarios para modificar el software. Este tipo de abuso sistemático ocurre con productos de uso personal, que es precisamente donde es menos aceptable. Por tanto, hemos diseñado esta versión de la GPL para prohibir estas prácticas en esos productos. Si apareciesen problemas similares en otros ámbitos, estaremos preparados para extender estas prestaciones a las próximas versiones de la GPL, tanto como sea necesario para proteger la libertad de los usuarios.

Por último, todo programa está constantemente amenazado por las patentes de software. Los estados no deberían permitir que las patentes restrinjan el desarrollo y el uso de software en ordenadores de uso general; pero en aquellos que lo hagan, esperamos evitar el especial peligro que suponen las patentes, que aplicadas a un programa libre puedan hacerlo propietario en la práctica. Para prevenir eso, la GPL establece que las patentes no pueden usarse para convertir un programa en no-libre.

A continuación se exponen los términos y condiciones para la copia, distribución y modificación.

A.2.3. TÉRMINOS Y CONDICIONES

Definiciones

En adelante “Esta Licencia” se refiere a la versión 3 de la Licencia Pública General GNU.

“Copyright” también significa “leyes similares al copyright” que son aplicables a otro tipo de trabajos, tales como las máscaras de semiconductores.

“El Programa” se refiere a cualquier trabajo con copyright al que se haya aplicado esta Licencia. Cada beneficiario es asimilable a “usted”. “Beneficiarios” y “destinatarios” pueden ser personas físicas u organizaciones.

“Modificar” un trabajo significa copiar o adaptar todo o parte de un trabajo, exceptuando la copia exacta, de manera que se requiera permiso de copyright. El trabajo resultante se denomina “versión modificada” de un trabajo anterior o trabajo “basado en” el trabajo anterior.

Un “trabajo amparado” puede ser tanto el Programa no modificado como un trabajo basado en el Programa.

“Difundir” un trabajo significa hacer cualquier cosa con él, sin permiso, que le haga directa o indirectamente responsable de infringir leyes cubiertas por copyright, excepto la ejecución en un ordenador o la modificación de una copia privada. La difusión incluye la copia, distribución (con o sin modificaciones), distribución pública, y en algunos países también otras actividades.

“Distribuir” un trabajo implica cualquier tipo de difusión que permite a la otra parte hacer o recibir copias. La mera interacción con un usuario mediante una red de ordenadores, sin transferir copia alguna, no se considera “distribución”.

Una interfaz de usuario interactiva muestra "Avisos Legales Apropiados" siempre y cuando incluya características visuales apropiadas y destacadas que (1) muestren un aviso de copyright apropiado, y (2) indiquen al usuario que no existe garantía alguna para el trabajo (exceptuando las garantías que se hayan podido establecer), que los beneficiarios deben distribuir el trabajo según se establece en la presente Licencia, y cómo se puede ver una copia de esta Licencia. Si la interfaz muestra una lista de opciones o comandos, tales como menús, un elemento destacado en dicha lista cumple estos criterios.

Código Fuente

El "código fuente" de un trabajo es el formato preferido para realizar modificaciones sobre él. "Código objeto" se refiere a cualquier formato del trabajo que no sea código fuente.

Una "Interfaz Estándar" se refiere a una interfaz que sea o bien un estándar oficial definido por una institución de estándares reconocida, o bien, en el caso de interfaces específicos para una determinado lenguaje de programación, una cuyo uso esté generalizada entre los desarrolladores que trabajan con ese lenguaje.

Las "Bibliotecas de Sistema" de un trabajo ejecutable incluyen a cualquier elemento, que no sea el trabajo completo, que (a) esté incluida/o de la misma forma que un componente principal, pero que no forme parte de ese componente principal, y (b) sólo sirva para habilitar la utilización del trabajo a través de ese componente principal, o para implementar un Interfaz Estándar para el cual está disponible una implementación pública en código fuente. Un "Componente Principal", en este contexto, se refiere a un componente principal y esencial (núcleo, sistema de ventanas y similares) del sistema operativo particular (en su caso) sobre el cual funcione el ejecutable, o un compilador utilizado para generar el trabajo, o un intérprete del código objeto utilizado para ejecutarlo.

La "Fuente Correspondiente" de un trabajo en código objeto se refiere a todo código fuente necesario para generar, instalar, y (en el caso de trabajos ejecutables) ejecutar el código objeto y modificar el trabajo, incluyendo guiones que controlen esas actividades. Sin embargo, no se incluyen las Bibliotecas de Sistema del trabajo, o herramientas de propósito general o programas gratuitos habitualmente disponibles y usados sin ninguna modificación para realizar estas actividades pero que no forman parte del trabajo. Por ejemplo, la Fuente Correspondiente incluye los archivos de definición de interfaz asociados con archivos fuente del trabajo, y el código fuente de las bibliotecas compartidas o subprogramas enlazados dinámicamente que el programa requiere por diseño, como la comunicación de datos intrínseca o el control de flujo entre esos subprogramas y otras partes del trabajo.

La Fuente Correspondiente no incluye necesariamente aquello que los usuarios pueden regenerar automáticamente a partir de otras partes de la Fuente Correspondiente.

La Fuente Correspondiente de un trabajo en código fuente es ese mismo trabajo.

Permisos Básicos

Todos los derechos garantizados por esta Licencia se otorgan como copyright del Programa, y se proporcionan de manera irrevocable siempre y cuando se cumplan las condiciones establecidas. Esta Licencia afirma explícitamente su permiso ilimitado para ejecutar el Programa sin modificaciones. El resultado de la ejecución de un programa amparado está cubierto por esta Licencia sólo en el caso de que la salida, por su contenido, constituya un trabajo amparado. Esta Licencia reconoce sus derechos de uso razonable u otro equivalente, tal y como determina la ley de copyright.

Usted podrá realizar, ejecutar y difundir trabajos amparados que no distribuya, sin condición alguna, siempre y cuando no tenga otra licencia más restrictiva. Podrá distribuir trabajos amparados a terceros con el mero objetivo de que ellos hagan modificaciones exclusivamente para usted, o para que le proporcionen ayuda para ejecutar esos trabajos, siempre que cumpla los términos de esta Licencia distribuyendo todo el material de cuyo copyright no posee el control. Aquellos que realicen o ejecuten los trabajos amparados para usted deben hacerlo exclusivamente en su nombre, bajo su dirección y control, con términos que les prohíban realizar copias de su material con copyright al margen de la relación con usted.

La distribución bajo otras circunstancias se permite únicamente bajo las condiciones establecidas más abajo. No está permitido sublicenciar; la cláusula 10 lo hace innecesario.

Protección de Derechos Legales de los Usuarios frente a Leyes Anti-Burla

Ningún trabajo amparado debe considerarse parte de una medida tecnológica efectiva, a tenor de lo establecido en cualquier ley aplicable que cumpla las obligaciones expresas en el artículo 11 del tratado de copyright WIPO adoptado el 20 de diciembre de 1996, o leyes similares que prohíben o restringen la burla de tales medidas.

Cuando distribuya un trabajo amparado, renuncia a cualquier poder legal para prohibir la burla de medidas tecnológicas mientras tales burlas se realicen en ejercicio de derechos amparados por esta Licencia respecto al trabajo amparado; además, usted negará cualquier intención de limitar el uso o modificación del trabajo con el objetivo de imponer, al trabajo de los usuarios, sus derechos legales o de terceros para prohibir la burla de medidas tecnológicas.

Distribución de copias literales

Usted podrá distribuir copias literales del código fuente del Programa tal y como lo ha recibido , por cualquier medio, siempre que publique de forma clara y llamativa en cada copia el correspondiente aviso de copyright ; mantenga intactos todos los avisos que establezcan que esta Licencia y cualquier término no-permisivo añadido y acorde con la cláusula 7 son aplicables al código; mantenga intactos todos los avisos de ausencia de garantía; y proporcione a todos los destinatarios una copia de esta Licencia junto con el Programa.

Usted podrá cobrar cualquier importe o no cobrar nada por cada copia que distribuya, y podrá ofrecer soporte o protección de garantía mediante un pago.

Distribución de Versiones Modificadas de Código

Usted podrá distribuir un trabajo basado en el Programa, o las modificaciones que lo producen a partir del Programa, como código fuente en virtud de los términos establecidos en la cláusula 4, siempre que cumpla todas las condiciones siguientes:

- El trabajo debe incluir avisos destacados indicando que usted lo ha modificado y dando una fecha pertinente.
- El trabajo debe incluir avisos destacados indicando que está realizado a tenor de lo dispuesto en la presente Licencia y en cualquier otra condición añadida en virtud de la cláusula 7. Este requisito modifica el requisito de "mantener intactos todos los avisos" expuesto en la cláusula 4.
- En virtud del presente documento, usted deberá aplicar la licencia al trabajo completo, como un todo, a cualquier persona que esté en posesión de una copia. Por lo tanto, esta Licencia se aplicará junto con cualquier otra condición adicional aplicable de la cláusula 7, al conjunto completo del trabajo y todas y cada una de sus partes, independientemente de cómo sean agrupadas o empaquetadas. Esta Licencia no permite ser aplicada al trabajo de ninguna otra forma, pero no se anula dicho permiso si usted lo ha recibido por separado.
- Si el trabajo tiene interfaces de usuario interactivos, cada uno debe mostrar Avisos Legales Apropiados; sin embargo, si el Programa tiene interfaces interactivos que no muestran Avisos Legales Apropiados, su trabajo no tiene porqué modificarlos para que lo hagan.

Un conjunto o recopilación formado por un trabajo amparado y otros trabajos distintos e independientes, que por su naturaleza no sean ampliaciones del trabajo amparado, que no se combinen con él de alguna forma para dar lugar a un programa mayor, y que estén ubicados en un medio de distribución o almacenamiento, se denomina "paquete" si la recopilación y su copyright al completo no son usados para limitar el acceso o los derechos legales de los usuarios de la recopilación, más allá de lo que permita el trabajo individual. La inclusión de un trabajo amparado en un paquete no hace aplicable esta Licencia al resto de elementos del paquete.

Distribución de código No-fuente

Usted podrá distribuir el código objeto de un trabajo amparado en virtud de los términos de las cláusulas 4 y 5, siempre que también distribuya las Fuentes Correspondientes en código máquina, de acuerdo con los términos establecidos en esta Licencia, de alguna de las siguientes maneras:

- Distribuir el código objeto en, o embebido en, un producto físico (incluyendo medios de distribución físicos), acompañado de las Fuentes Correspondientes en un medio físico duradero y que sea utilizado habitualmente para el intercambio de software.

- Distribuir el código objeto en, o embebido en, un producto físico (incluyendo medios de distribución físicos), acompañado de una oferta por escrito, válida al menos durante tres años y válida durante el tiempo en el que usted ofrezca recambios o soporte para ese modelo de producto, con el fin de ofrecer al poseedor del código objeto (1) una copia de las Fuentes Correspondientes a todo el software del producto que esté cubierto por esta Licencia, en un medio físico duradero habitual para el intercambio de software, a un precio no mayor que su coste razonable por distribuir físicamente las fuentes, o (2) acceso para copiar las fuentes correspondientes desde un servidor de red sin coste alguno.

- Distribuir copias individuales del código objeto junto con una copia de la oferta por escrito para/con el fin de proporcionar las Fuentes Correspondientes. Esta alternativa sólo está permitida ocasionalmente, pero no de forma comercial, y solamente si usted recibió el código objeto junto con una oferta parecida, de acuerdo con la subcláusula 6b.

- Distribuir el código objeto ofreciendo acceso desde un lugar determinado (gratuitamente o mediante pago), y ofrecer acceso equivalente a las Fuentes Correspondientes de la misma forma y en el mismo lugar sin cargo añadido. No es necesario exigir a los destinatarios que copien las Fuentes Correspondientes junto con el código objeto. Si el lugar para copiar el código objeto es un servidor de red, las Fuentes Correspondientes pueden estar en un servidor diferente (gestionado por usted o terceros) que ofrezca facilidades de copia equivalentes, siempre que mantenga instrucciones claras junto al código objeto especificando dónde encontrar las Fuentes Correspondientes. Independientemente de qué servidores alberguen las Fuentes Correspondientes, usted seguirá obligado a asegurar que estarán disponibles durante el tiempo necesario para cumplir estos requisitos.

- Distribuir el código mediante transferencias entre usuarios, siempre que informe a otros usuarios dónde se ofrecen el código objeto y las Fuentes Correspondientes de forma pública sin cargo alguno, tal y como se establece en la subcláusula 6d.

Una parte separable del código objeto, cuyo código fuente esté excluido de las Fuentes Correspondientes como Biblioteca de Sistema, no necesita ser incluida en la distribución del código objeto del trabajo.

Un "Producto de Usuario" es tanto (1) un "producto de consumo", que se refiere a cualquier propiedad personal tangible habitualmente utilizada para fines personales, familiares o domésticos, o (2) cualquier cosa diseñada o vendida para ser incorporada como extensión/expansión para otro producto. Para determinar si un producto es un producto de consumo, los casos dudosos se resolverán favoreciendo el amparo. En el caso de un producto concreto recibido por un usuario particular, "de uso habitual" se refiere al uso típico o corriente de ese

tipo de producto, independientemente de la situación del usuario particular o de la forma en que el usuario concreto utilice, o pretenda o se espere que pretenda utilizar, el producto. Un producto es un producto de consumo independientemente de si el producto tiene usos sustancialmente comerciales, industriales o distintos del consumo, a menos que tales usos representen la única forma posible de utilizar el producto.

Las "Instrucciones de Instalación" para un Producto de Usuario se refieren a cualquier método, procedimiento, clave de autorización, u otro tipo de información necesaria para instalar y ejecutar una versión modificada de un trabajo amparado en ese Producto de Usuario a partir de una versión modificada de las Fuentes Correspondientes. Las instrucciones deben ser suficientes para asegurar el funcionamiento continuo del código objeto modificado sin ningún tipo de condicionamiento o intromisión por el simple hecho de haber sido modificado.

Si, bajo las premisas de esta cláusula, usted distribuye el código objeto de un trabajo en, o con un Producto de Usuario o específicamente para ser usado en el mismo, y la distribución forma parte de una transacción donde los derechos de posesión y uso del Producto de Usuario se transfieren al destinatario a perpetuidad o durante un plazo fijo de tiempo (independientemente de las características de la transacción), las Fuentes Correspondientes distribuidas bajo estos supuestos deben acompañarse de las Instrucciones de Instalación. Sin embargo, estos requerimientos no se aplican si ni usted ni terceros tienen posibilidad de instalar el código objeto modificado en el Producto de Usuario (por ejemplo, el trabajo ha sido instalado en memoria de sólo lectura, ROM):

El requerimiento de proporcionar Información de Instalación no incluye el hecho de continuar proporcionando servicio de soporte, garantía, o actualizaciones para un trabajo que haya sido modificado o instalado por el destinatario, o para el Producto de Usuario en el que se haya modificado o instalado. El acceso a la red puede ser denegado cuando la propia modificación afecte materialmente y de forma adversa a la operación de la red o viole las reglas y protocolos de comunicación en la red.

Las Fuentes Correspondientes distribuidas, y las Instrucciones de Instalación proporcionadas de acuerdo con esta cláusula, deben figurar en un formato documentado públicamente (y con una implementación disponible para el público en código fuente), y no deben necesitar claves de acceso especiales para la descompresión, lectura o copia.

Condiciones adicionales

Los "Permisos Adicionales" son condicionantes que amplían los términos de esta Licencia permitiendo excepciones a una o más de sus condiciones. Los Permisos Adicionales que son aplicables al Programa completo deberán ser tratados como si estuviesen incluidos en esta Licencia, hasta los límites de validez impuestos por las leyes aplicables. Si los permisos adicionales se aplicasen sólo a una parte del Programa, esa parte podría ser usada de forma independiente en virtud de dichos permisos, pero el Programa completo seguiría estando afectado por esta Licencia con independencia de los permisos adicionales.

Cuando distribuya una copia de un trabajo amparado, usted podrá opcionalmente eliminar cualquier permiso adicional de esa copia, o de alguna parte del

mismo. (Los permisos adicionales pueden haber establecido que sea requerida su eliminación en ciertos supuestos si usted modifica el trabajo.) Usted puede establecer permisos adicionales en material añadido por usted a un trabajo amparado, sobre el cual tiene o podrá aportar sus permisos de copyright correspondientes.

Sin contravenir cualquier otra estipulación en esta Licencia, usted podrá, para el material que añada a un trabajo amparado, (si está autorizado por los poseedores de copyright de ese material) añadir condiciones a esta Licencia con los siguientes términos:

- Ausencia de garantía o limitación de responsabilidad diferente de los términos establecidos en las cláusulas 15 y 16 de esta Licencia.
- Obligación de mantener determinados avisos legales razonables o atribuciones de autoría en el material o en los Avisos Legales Correspondientes mostrados por los trabajos que lo contengan.
- Prohibir la tergiversación del origen del material, o solicitar que las diferencias respecto a la versión original sean señaladas de forma apropiada en las versiones modificadas del material.
- Limitar la utilización de los nombres de los autores o beneficiarios del material con fines divulgativos.
- Negarse a ofrecer derechos afectados por leyes de registro para el uso de marcas empresariales, registradas o de servicio.
- Exigir indemnización a los autores y poseedores de la licencia de ese material, por parte de cualquier persona que distribuya el material (o versiones modificadas del mismo), estableciendo obligaciones contractuales de responsabilidad sobre el destinatario, para cualquier responsabilidad que estas obligaciones contractuales impongan directamente sobre los autores y poseedores de licencia.

Cualesquiera otras condiciones adicionales no-permisivas son consideradas "otras restricciones" en el contexto de la cláusula 10. Si el Programa, tal cual lo recibió, o cualquier parte del mismo, contiene un aviso indicando que está amparado por esta Licencia junto a una cláusula de restricción posterior específica, usted podrá suprimir esa cláusula. Si un documento de licencia contiene una restricción de este tipo pero permite modificar la licencia o la distribución en virtud de la presente Licencia, usted podrá añadirla al material de un trabajo amparado por los términos de ese documento de licencia, siempre que dicha restricción no se mantenga tras la modificación de la licencia o la distribución.

Si añade condiciones para un trabajo amparado, a tenor de lo establecido en la presente cláusula, usted deberá ubicar, en los archivos fuente involucrados, una declaración de los términos adicionales aplicables a esos archivos, o un aviso indicando dónde localizar los términos aplicables.

Las condiciones adicionales, permisivas o no, deben aparecer por escrito como licencias separadas, o figurar como excepciones; de todas formas, los requisitos anteriores siempre son aplicables.

Cancelación

Usted no podrá distribuir o modificar un trabajo amparado salvo de la forma en la que se ha previsto expresamente en esta Licencia. Cualquier intento diferente de distribución o modificación será considerado nulo, y automáticamente cancelará sus derechos respecto a esta Licencia (incluyendo cualquier patente conseguida según el párrafo tercero de la cláusula 11).

Sin embargo, si deja de violar esta Licencia, entonces su licencia desde el poseedor del copyright correspondiente será restituida (a) provisionalmente, a menos que y hasta que el poseedor del copyright dé por terminada explícita y permanentemente su licencia, y (b) permanentemente, si el poseedor del copyright no le ha notificado por algún cauce de la violación no después de los 60 días posteriores al cese.

Además, su licencia desde el poseedor del copyright correspondiente será restituida permanentemente si el poseedor del copyright le notifica de la violación por algún cauce, es la primera vez que recibe la notificación de violación de esta Licencia (para cualquier trabajo) de ese poseedor de copyright, y usted subsana la violación antes de 30 días desde la recepción del aviso.

La cancelación de sus derechos según esta cláusula no da por canceladas las licencias de terceros que hayan recibido copias o derechos a través de usted con esta Licencia. Si sus derechos han finalizado y no han sido restituidos de forma permanente, usted no está capacitado para recibir nuevas licencias para el mismo material en virtud de la cláusula 10.

Aceptación no obligatoria por tenencia de copias

No está obligado a aceptar esta Licencia por recibir o ejecutar una copia del Programa. La distribución de un trabajo amparado surgida simplemente como consecuencia de la transmisión entre usuarios para obtener una copia tampoco requiere aceptación. Sin embargo, únicamente esta Licencia le otorga permiso para distribuir o modificar cualquier trabajo amparado. Estas acciones infringen el copyright si usted no acepta las los términos y condiciones de esta Licencia. Por lo tanto, al modificar o distribuir un trabajo amparado, usted indica que acepta la Licencia.

Herencia automática de licencia para destinatarios

Cada vez que distribuya un trabajo amparado, el destinatario recibirá automáticamente una licencia desde los poseedores originales, para ejecutar, modificar y distribuir ese trabajo, al amparo de los términos de esta Licencia. Usted no será responsable de asegurar el cumplimiento por terceros de esta Licencia.

Una “transacción de entidad” es una transacción que transfiere el control de una organización, o todos los bienes sustanciales de una, o subdivide una organización, o fusiona organizaciones. Si la distribución de un trabajo amparado surge de una transacción de entidad, cada parte involucrada en esa transacción que reciba una copia del trabajo, también recibe todas y cada una de las licencias existentes del trabajo que la parte interesada tuviese o pudiese ofrecer según el párrafo anterior, además del derecho a tomar posesión de las Fuentes

Correspondientes del trabajo a través de la parte interesada, si está en poder de dicha parte o se puede conseguir con un esfuerzo razonable.

Usted no podrá imponer restricciones posteriores en el ejercicio de los derechos otorgados o concedidos en virtud de la presente Licencia. Por ejemplo, usted no puede imponer a la licencia pagos, derechos u otros cargos por el ejercicio de los derechos otorgados según esta Licencia; además no podrá iniciar litigios (incluyendo demandas o contrademandas en pleitos) alegando que se infringen patentes por cambiar, usar, vender, ofrecer en venta o importar el Programa, o cualquier parte del mismo.

Patentes

Un "colaborador" es un poseedor de copyright que autoriza el uso del Programa o un trabajo en el que se base el Programa bajo los términos y condiciones establecidos en la presente Licencia. El trabajo con esta licencia se denomina "versión en colaboración" con el colaborador.

Todas las reivindicaciones de patentes en posesión o controladas por el colaborador se denominan "demandas de patente original", ya sean existentes o adquiridas con posterioridad, que hayan sido infringidas de alguna forma permitida por esta Licencia, al hacer, usar o vender la versión en colaboración, pero sin incluir demandas que sólo sean infracciones como consecuencia de modificaciones posteriores de la versión en colaboración. Para aclarar esta definición, "control" incluye el derecho de conceder sublicencias de patente de forma que no contravenga los requisitos establecidos en la presente Licencia.

Cada colaborador le concede a usted una licencia de la patente no-exclusiva, global y libre de derechos bajo las reivindicaciones de patente de origen del colaborador, para el uso, modificación, venta, ofertas de venta, importación y otras formas de ejecución, modificación y redistribución del contenido de la versión en colaboración.

En los siguientes tres párrafos, una "licencia de patente" se refiere a cualquier acuerdo o compromiso expreso y manifiesto, cualquiera que sea su denominación, que no imponga una patente (como puede ser el permiso expreso para ejecutar una patente o acuerdos para no imponer demandas por infracción de patente). "Conceder" estas licencias de patente a un tercero significa llegar a tal tipo de acuerdo o compromiso que no imponga una patente al tercero.

Si usted distribuye un trabajo amparado, conociendo que está afectado por una licencia de patente, y no están disponibles de forma pública para su copia las Fuentes Correspondientes, sin cargo alguno y bajo los términos de esta Licencia, ya sea a través de un servidor de red público o mediante cualquier otro medio, entonces usted deberá o bien (1) permitir que sean públicas las Fuentes Correspondientes, o (2) tratar de eliminar los beneficios de la licencia de patente para este trabajo en particular, o (3) tratar de extender, de una forma que no contravenga los requisitos de esta Licencia, la licencia de patente a terceros. "Conocer que está afectado" significa que usted tiene conocimiento real de que, para la licencia de patente, la distribución del trabajo amparado en un determinado país, o el uso del trabajo amparado por sus destinatarios en un determinado país, infringiría una o más patentes existentes en ese país que usted considera aplicables por algún motivo.

Si, de conformidad con alguna transacción o acuerdo(o en un proceso relacionado con ellos), usted distribuye o distribuye con fines de distribución , un trabajo amparado, concediendo una licencia de patente para algún tercero que reciba el trabajo amparado, y autorizándole a usar, distribuir, modificar o distribuir una copia específica del trabajo amparado, entonces la licencia de patente que usted otorgue se extiende automáticamente a todos los receptores del trabajo amparado y cualquier trabajo basado en el mismo.

Una licencia de patente es "discriminatoria" si no incluye dentro de su ámbito de cobertura, prohíbe el ejercicio, o está condicionada a no ejercitar uno o más de los derechos que están específicamente otorgados por esta Licencia. Usted no debe distribuir un trabajo amparado si está implicado en un acuerdo con terceros que estén relacionados con el negocio de la distribución de software, en el que usted haga pagos relacionados con su actividad de distribución del trabajo, y donde se otorgue, a cualquier receptor del trabajo amparado, una licencia de patente discriminatoria (a) en relación con las copias del trabajo amparado distribuido por usted (o copias hechas a partir de éstas), o (b) directa o indirectamente relacionadas con productos específicos o paquetes que contengan el trabajo amparado, a menos que usted forme parte del acuerdo, o que esa licencia de patente fuese otorgada antes del 28 de marzo de 2007.

Ninguna disposición de esta Licencia se considerará como excluyente o limitante de la aplicación de cualquier otra licencia o defensas legales contra la violación de las leyes de propiedad intelectual a que pudiera tener derecho bajo la ley de propiedad intelectual vigente.

No condicionamiento de la libertad de terceros

Si a usted le son impuestas condiciones que contravienen las estipuladas en la presente Licencia (ya sea por orden judicial, acuerdo u otros), no quedará eximido de cumplir las condiciones de esta Licencia. Si usted no puede distribuir un trabajo amparado cumpliendo simultáneamente sus obligaciones con esta Licencia y con cualquier otra pertinente, entonces no podrá distribuirlo de ninguna forma. Por ejemplo, si usted se compromete con términos que le obligan a obtener derechos por la distribución a terceros, la única forma de satisfacer ambos condicionantes y esta Licencia es abstenerse completamente de distribuir el Programa.

Uso conjunto con la Licencia Pública General Affero GNU

Sin contravenir las disposiciones de la presente Licencia, usted tendrá permiso para enlazar o combinar cualquier trabajo amparado con otro trabajo amparado por la versión 3 de la Licencia Pública General Affero GNU y formar un solo trabajo combinado, y distribuir el trabajo resultante. Los términos de esta Licencia seguirán siendo aplicables a la parte formada por el trabajo amparado, pero los condicionantes especiales de la Licencia Pública General Affero GNU, en su cláusula 13, relativos a la interacción mediante redes, serán aplicables a la combinación de ambas partes.

Versiones Revisadas de esta Licencia

La Fundación para el Software Libre podrá publicar revisiones y/o nuevas versiones de la Licencia Pública General GNU de vez en cuando. Esas versiones serán similares en espíritu a la versión actual, pero podrán diferir en algunos detalles para afrontar nuevos problemas o situaciones.

A cada versión se le da un número distintivo. Si el Programa especifica que le es aplicable cierto número de versión de la Licencia Pública General o "cualquier versión posterior", usted tendrá la posibilidad de adoptar los términos y condiciones de la versión indicada o de cualquier otra versión posterior publicada por la Fundación para el Software Libre. Si el Programa no especifica un número de versión de la Licencia Pública General, usted podrá elegir cualquier versión que haya sido publicada por la Fundación para el Software Libre.

Si el Programa especifica que un apoderado/representante puede decidir qué versiones de la Licencia Pública General pueden aplicarse en el futuro, la declaración pública de aceptación que el apoderado/representante haga de una versión le autoriza a usted con carácter permanente a elegir esa versión para el Programa.

Versiones posteriores de la licencia podrán otorgarle permisos adicionales o diferentes. Sin embargo, no podrán imponerse obligaciones adicionales a cualquier autor o poseedor de copyright como consecuencia de que usted adopte una versión posterior.

Ausencia de Garantía

EL PROGRAMA NO TIENE GARANTÍA ALGUNA, HASTA LOS LÍMITES PERMITIDOS POR LAS LEYES APLICABLES. SALVO CUANDO SE ESTABLEZCA LO CONTRARIO POR ESCRITO, EL POSEEDOR DEL COPYRIGHT Y/O TERCEROS PROPORCIONARÁN EL PROGRAMA "TAL CUAL"SIN GARANTÍA DE NINGÚN TIPO, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS MERCANTILES Y DE APTITUD PARA UN PROPÓSITO DETERMINADO. USTED ASUMIRÁ CUALQUIER RIESGO RELATIVO A LA CALIDAD Y RENDIMIENTO DEL PROGRAMA. SI EL PROGRAMA FUESE DEFECTUOSO, USTED ASUMIRÁ CUALQUIER COSTE DE SERVICIO, REPARACIÓN O CORRECCIÓN.

Limitación de Responsabilidad

EN NINGÚN CASO, SALVO REQUERIMIENTO POR LEYES APLICABLES O MEDIANTE ACUERDO POR ESCRITO, PODRÁ UN POSEEDOR DE COPYRIGHT, O UN TERCERO QUE MODIFIQUE O DISTRIBUYA EL PROGRAMA SEGÚN LO INDICADO ANTERIORMENTE, HACERLE A USTED RESPONSABLE DE DAÑO ALGUNO, INCLUYENDO CUALQUIER DAÑO GENERAL, ESPECIAL, OCASIONAL O DERIVADO QUE SURJA DEL USO O LA INCAPACIDAD DE USO DEL PROGRAMA (INCLUYENDO PERO SIN LIMITARSE A LA PÉRDIDA DE DATOS O LA PRESENTACIÓN NO PRECISA DE LOS MISMOS O A PÉRDIDAS SUFRIDAS POR USTED O TERCEROS O AL FALLO DEL PROGRAMA AL INTERACTUAR

CON OTROS PROGRAMAS), INCLUSO EN EL CASO DE QUE EL POSEEDOR O UN TERCERO HAYA SIDO ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS.

Interpretación de las cláusulas 15 y 16

Si la ausencia de garantía y la limitación de responsabilidad descrita anteriormente no tuviesen efecto legal a nivel local en todos sus términos, los juzgados aplicarán las leyes locales que más se aproximen a la exención de responsabilidad civil en lo relativo al Programa, a menos que la copia del Programa esté acompañada mediante pago de una garantía o compromiso de responsabilidad.

FIN DE TÉRMINOS Y CONDICIONES

Cómo aplicar estas condiciones a sus nuevos programas

Si usted desarrolla un nuevo programa, y quiere darle al público el mayor uso posible del mismo, la mejor forma de conseguirlo es hacerlo software libre para que cualquiera pueda redistribuirlo y modificarlo bajo estas condiciones.

Para ello, adjunte los siguientes avisos al programa. Es más seguro adjuntarlos al inicio de cada archivo fuente para hacer más explícita la ausencia de garantía; y cada archivo debería tener al menos la línea de "copyright" y un enlace a la versión completa del aviso.

```
<una línea con el nombre del programa y una breve idea de su objetivo.>
Copyright (C) <año>  <nombre del autor>

Este programa es software libre: usted puede redistribuirlo y/o modificarlo
bajo los términos de la Licencia Pública General GNU publicada
por la Fundación para el Software Libre, ya sea la versión 3
de la Licencia, o (a su elección) cualquier versión posterior.

Este programa se distribuye con la esperanza de que sea útil, pero
SIN GARANTÍA ALGUNA; ni siquiera la garantía implícita
MERCANTIL o de APTITUD PARA UN PROPÓSITO DETERMINADO.
Consulte los detalles de la Licencia Pública General GNU para obtener
una información más detallada.

Debería haber recibido una copia de la Licencia Pública General GNU
junto a este programa.
En caso contrario, consulte <http://www.gnu.org/licenses/>.
```

Incluya además información de cómo contactar con usted por correo electrónico y ordinario.

Si el programa es interactivo, haga que muestre un breve aviso como el siguiente cuando se inicie en modo interactivo:

```
<programa>  Copyright (C) <año>  <nombre del autor>
Este programa se ofrece SIN GARANTÍA ALGUNA;
```

```
escriba 'show w' para consultar los detalles.
Es software libre, y usted puede redistribuirlo bajo ciertas condiciones;
escriba 'show c' para más información.
```

Los hipotéticos comandos 'show w' y 'show c' deberían mostrar las partes correspondientes de la Licencia Pública General. Por supuesto, los comandos en su programa podrían ser diferentes; en un interfaz gráfico de usuario, podría usar un mensaje del tipo "Acerca de".

También debería conseguir que su empresa (si trabaja como programador) o escuela, en su caso, firme una "renuncia de copyright" sobre el programa, si fuese necesario. Para más información a este respecto, y saber cómo aplicar y cumplir la licencia GNU GPL, consulte ¡http://www.gnu.org/licenses/¿.

La Licencia Pública General GNU no permite incorporar sus programas como parte de programas propietarios. Si su programa es una subrutina en una biblioteca, resultaría mucho más útil habilitar el enlace de aplicaciones propietarias a la biblioteca. Si es esto lo que quiere hacer, utilice la Licencia Pública General Reducida GNU en vez de esta Licencia. Pero por favor, consulte primero ¡http://www.gnu.org/philosophy/why-not-lgpl.html¿.